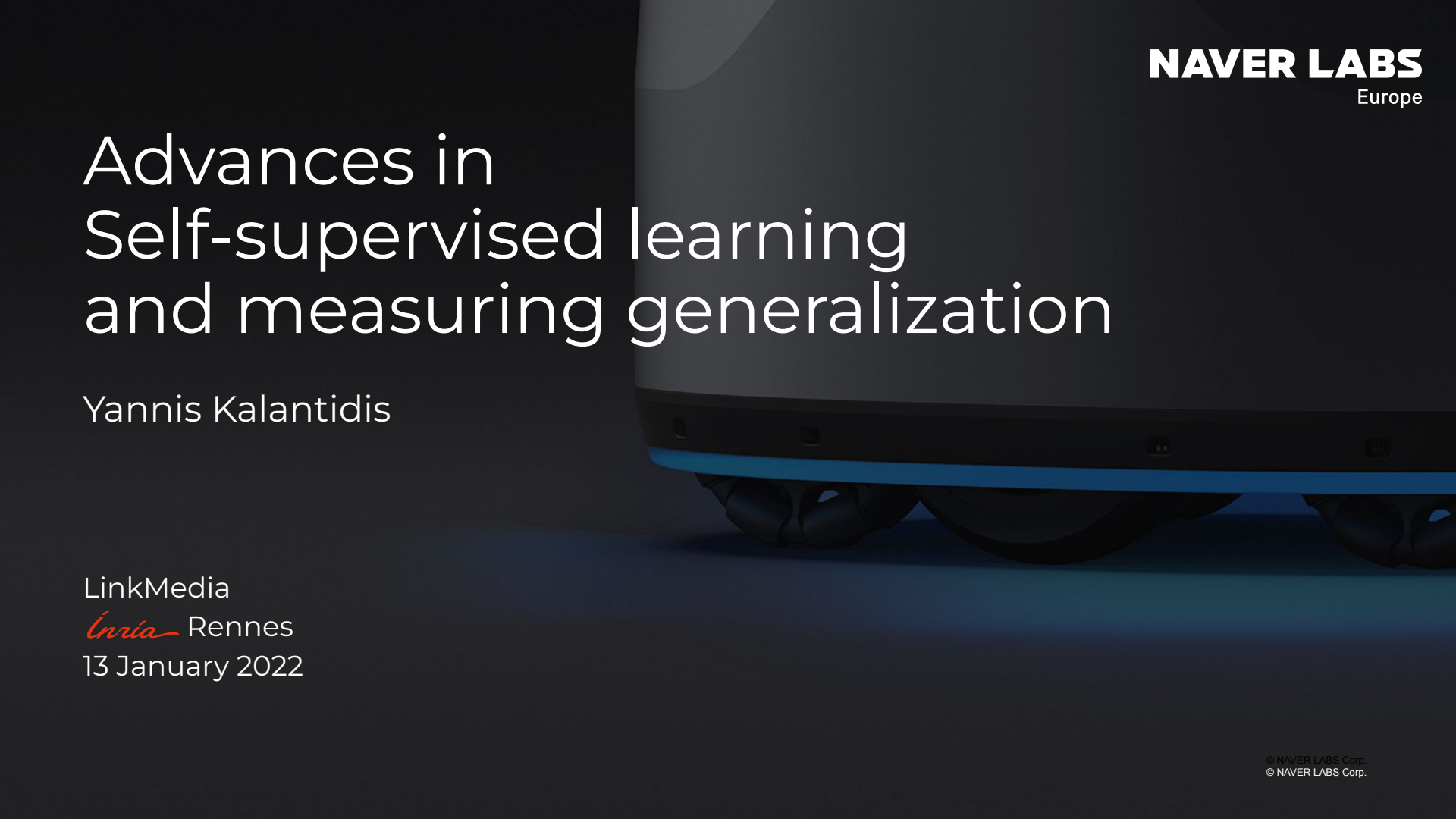


NAVER LABS
Europe

Advances in Self-supervised learning and measuring generalization

Yannis Kalantidis

LinkMedia
Inria Rennes
13 January 2022

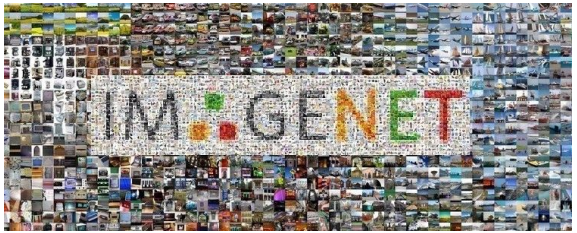
© NAVER LABS Corp.

Transfer learning

Transfer learning

a model developed for a task is reused as the starting point for a model on a second task

Model



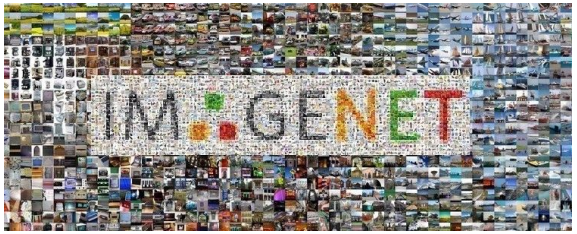
images and concept labels

Transfer learning

Supervised learning

Learn model parameters on a **labeled** dataset for image classification

Model



images and concept labels

"transfer" / "generalize to..."

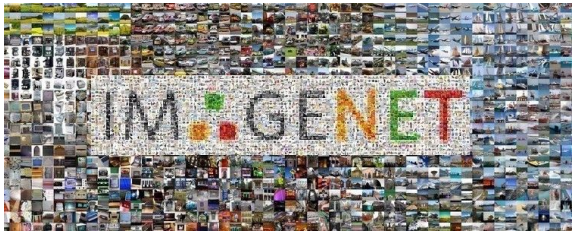
Downstream tasks & datasets

- Image Classification
- Object Detection
- Instance Segmentation
- Image Retrieval

Transfer learning

Can we learn transferable representations without requiring annotations?

Model



only images

Downstream tasks & datasets

Image Classification

Object Detection

Instance Segmentation

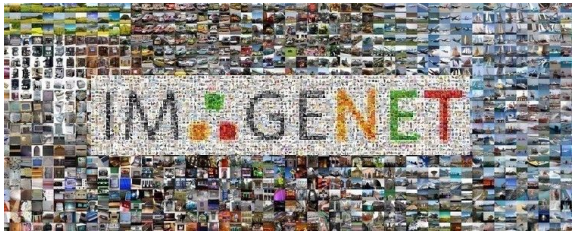
Image Retrieval

Transfer learning

Self-Supervised Learning

Learn model parameters
on an **unlabeled** dataset

Model



only images

Downstream tasks & datasets

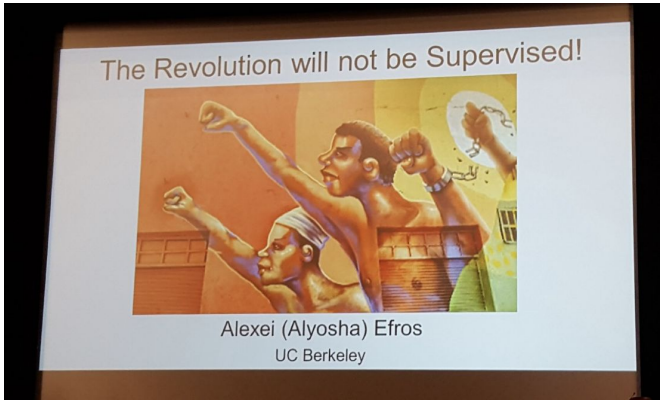
- Image Classification
- Object Detection
- Instance Segmentation
- Image Retrieval

Self-supervised learning (SSL)

Has deeply impacted the field of AI:

- Enables utilizing **unlabeled data**
- Revolutionized NLP (BERT/GPT-3 etc)
- Becoming a core component of computer vision state-of-the-art

Yann LeCun's talks (NeurIPS 2016 and many after)



Alyosha Efros' talks (ICCV 2017 and many after)

Self-supervised learning (SSL)

- Learn a self-supervised **proxy task**
  - A task defined on the input data alone
  - Learn “aspects” of the input

Self-supervised learning (SSL)

- Learn a self-supervised **proxy task**
  - A task defined on the input data alone
  - Learn “aspects” of the input
- ***No annotations*** *required!*
  - **Scalability**: use “any” image/video
  - **Flexibility**: find the data that fits your downstream task

Self-supervised learning (SSL)

- Learn
  - A
  - ne
- **No**
  - eo
  - **Flexi** a that fits
    downstream task

Does this mean that I don't need to care about what data I use anymore?

Self-supervised learning (SSL)

- Learn
  - A
  - ne
- **No**
  - eo
  - **Flexi** a that fits
    downstream task

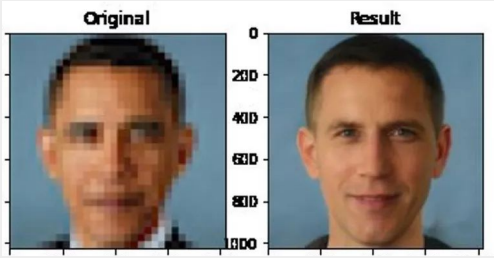
Does this mean that I don't need to care about what data I use anymore?

Nah-huh!!!
You need to care **even more!**

Self-supervised learning (SSL)

- Learn
  - A
  -
- **No**
  -
  - **Flexi** a that fits
    r downstream task

Does this mean that I don't need to care about what data I use anymore?

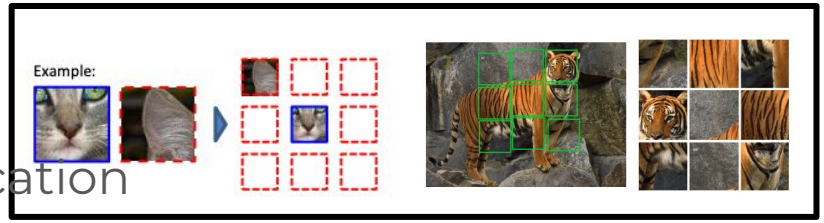


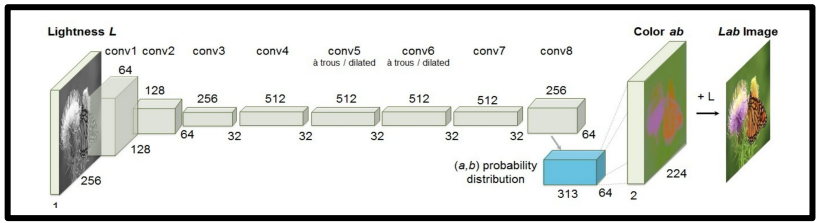
[Image Source: @Chicken3gg]

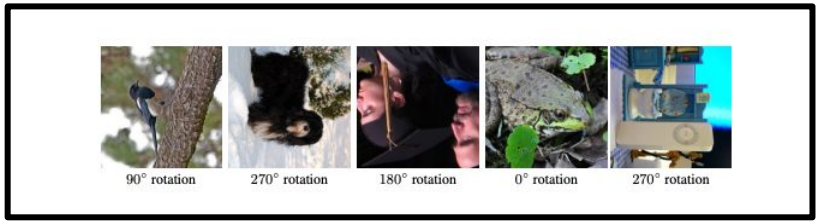
Self-supervised proxy task

Predictive/Generative

- Formulated as synthesis or classification







Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. **Unsupervised visual representation learning by context prediction.** ICCV. 2015.
Noroozi, Mehdi, and Paolo Favaro. **Unsupervised learning of visual representations by solving jigsaw puzzles** ECCV 2016.
Zhang, R., Isola, P., & Efros, A. A. **Colorful image colorization.** ECCV 2016.
Gidaris, S., Singh, P., & Komodakis, N. (2018). **Unsupervised representation learning by predicting image rotations.** ICLR 2018

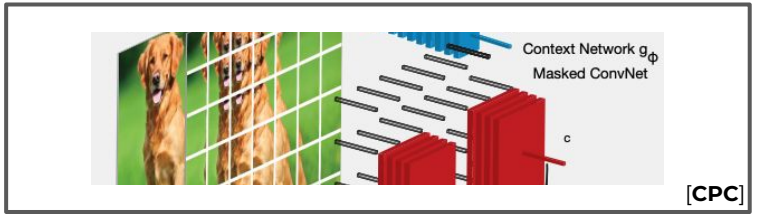
Self-supervised proxy task

Predictive/Generative

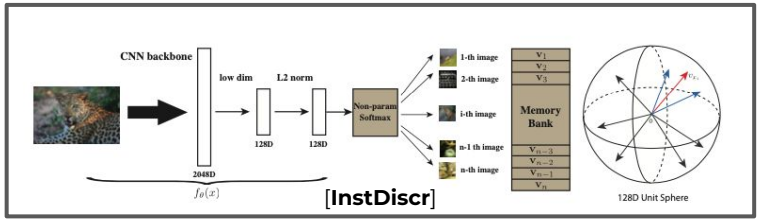
- Formulated as synthesis or classification

Contrastive

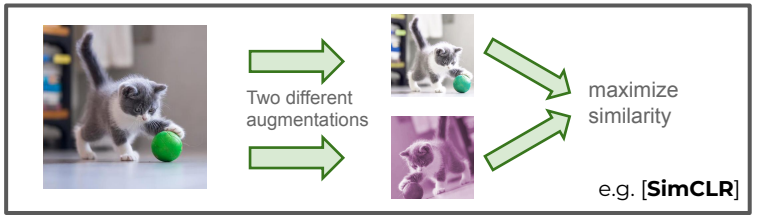
- Learning invariance to a “pretext” task



[**CPC**]



[**InstDiscr**]



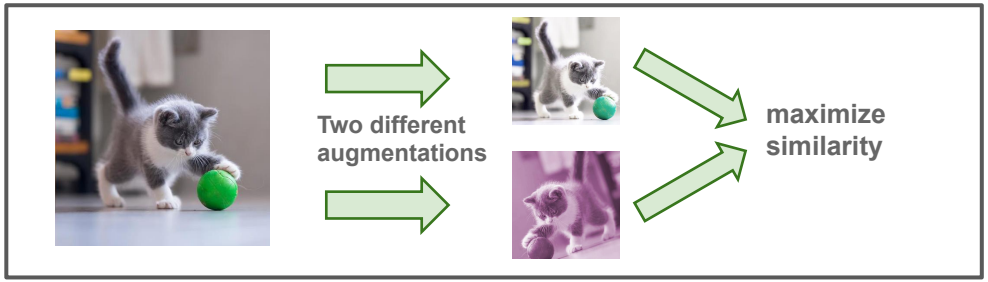
e.g. [**SimCLR**]

[**CPC**] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[**InstDiscr**] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.
[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020]

A very successful proxy task for visual representations

NAVER LABS

Learning invariance to image transformations (data augmentation)



State-of-the-art for learning generalizable **visual** representations

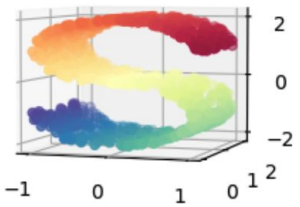
[Too-many-to-fit references - see Appendix]

Beyond pixel inputs

A more generic task
(independent of the specific nature of the input space):

Learn a low-dimensional space that preserves the topology of the input space

(a.k.a. Dimensionality reduction
or manifold learning)



Overview of this talk

Part 1

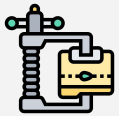
How can we improve the transfer learning performance of contrastive SSL?

[MoCHi]
Neurips 2020

Part 2

Can we use recent visual SSL frameworks for dimensionality reduction?

[TLDR]
Arxiv 2021



Part 3

How can we measure concept generalization in a more principled way?

[ImageNet-CoG]
ICCV 2021

[**MoCHi**] Kalantidis *et al.* "Hard negative mixing for contrastive learning." NeurIPS 2020.
[**TLDR**] Kalantidis *et al.* "TLDR: Twin Learning for dimensionality reduction" arXiv 2021.
[**ImageNet-CoG**] Sariyildiz, Kalantidis *et al.* "Concept Generalization in Visual Representation Learning" ICCV 2021.

icons from Flaticon.com

Part 1:
Improving contrastive self-supervised learning with hard negative mixing

NAVER LABS

Contrastive self-supervised learning

Contrastive self-supervised learning

- **Proxy task:**
  Learning invariance to image transformations (data augmentation)
- Define positive pairs and negatives in a **self-supervised way**
  - Positive pair: Two transformed versions of the same image

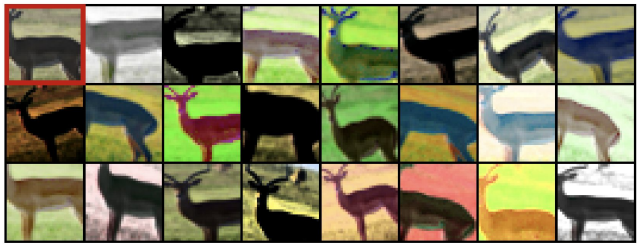


Figure from [**Exemplar-CNN**]

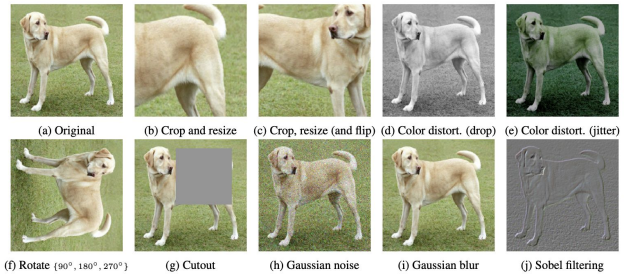


Figure from [**SimCLR**]

[**Exemplar-CNN**] Dosovitskiy, et al. "Discriminative unsupervised feature learning with exemplar convolutional neural networks." TPAMI 2015]
[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020]

Contrastive self-supervised learning

Contrastive self-supervised learning

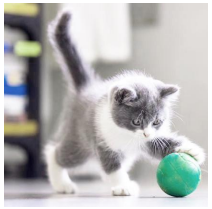
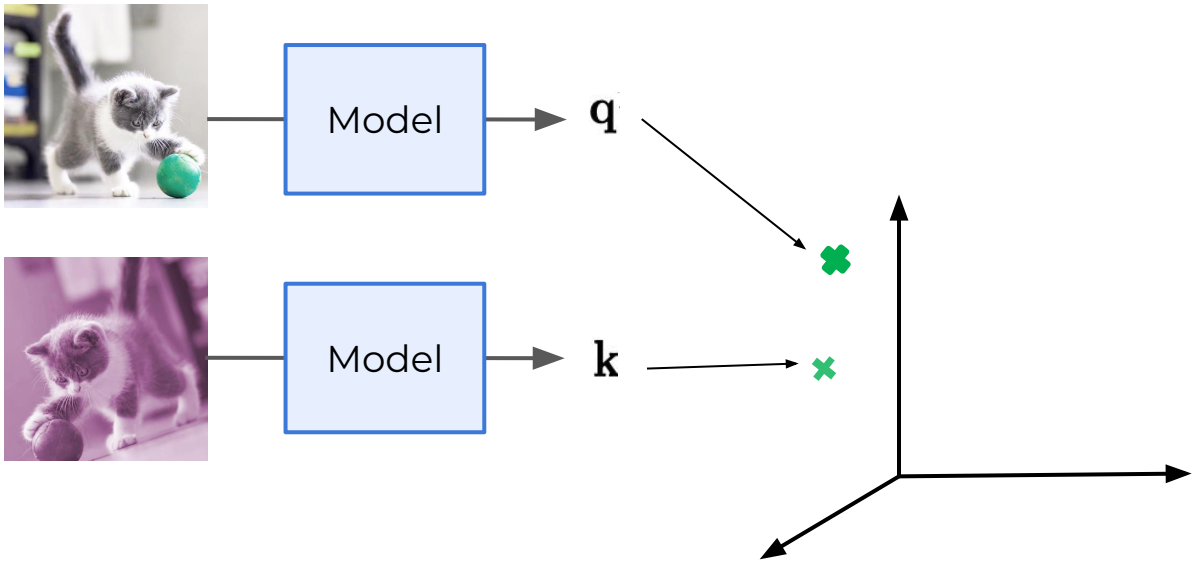
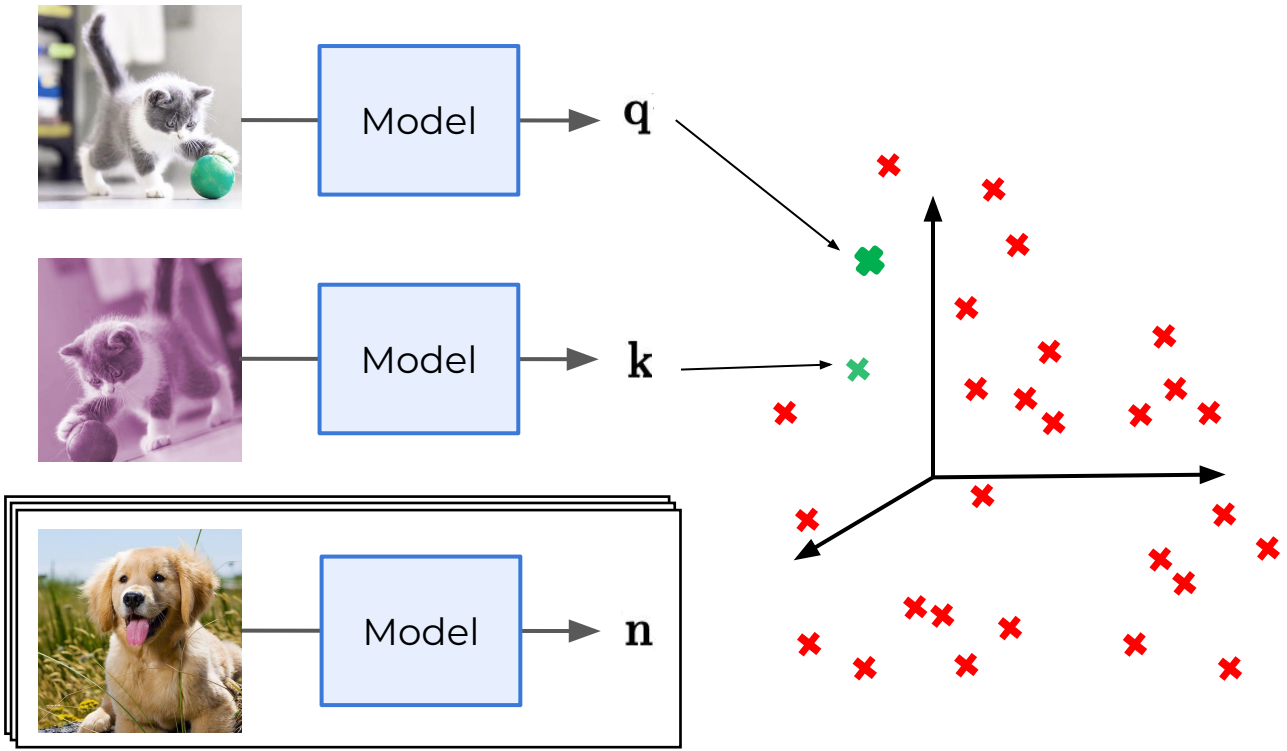


Image Transformations

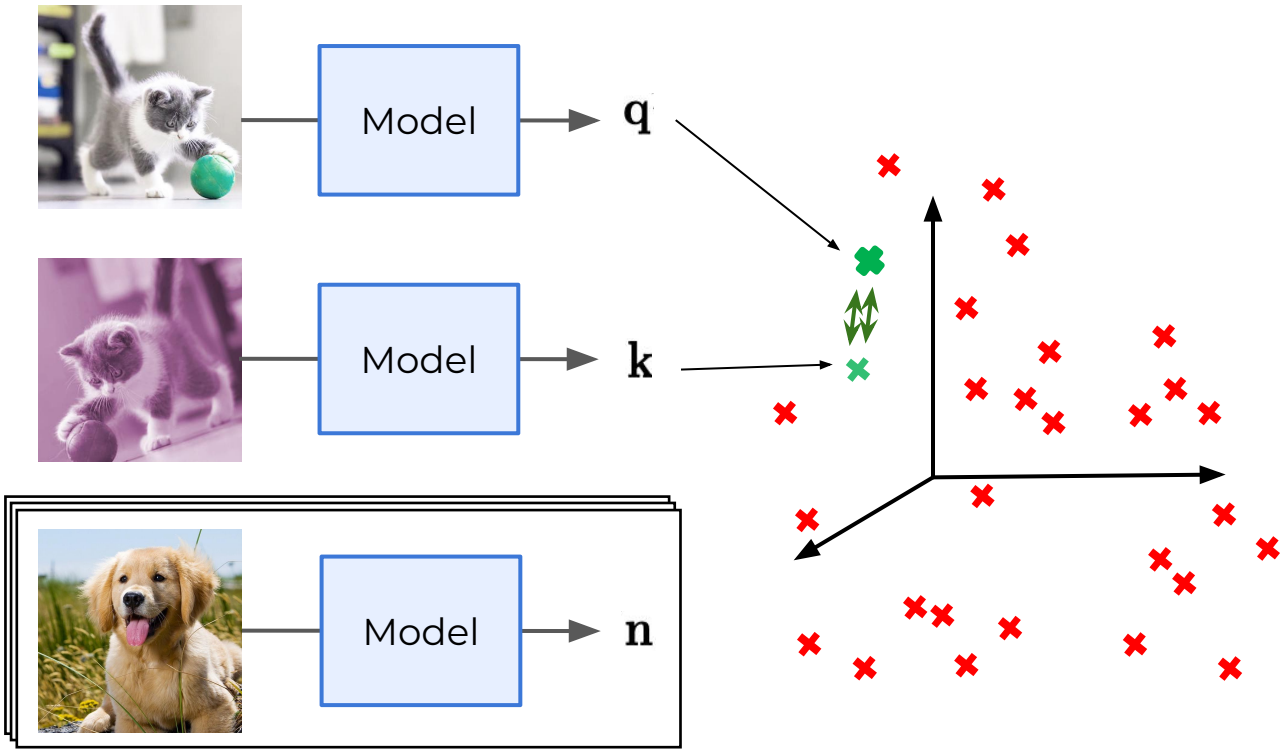
Contrastive self-supervised learning



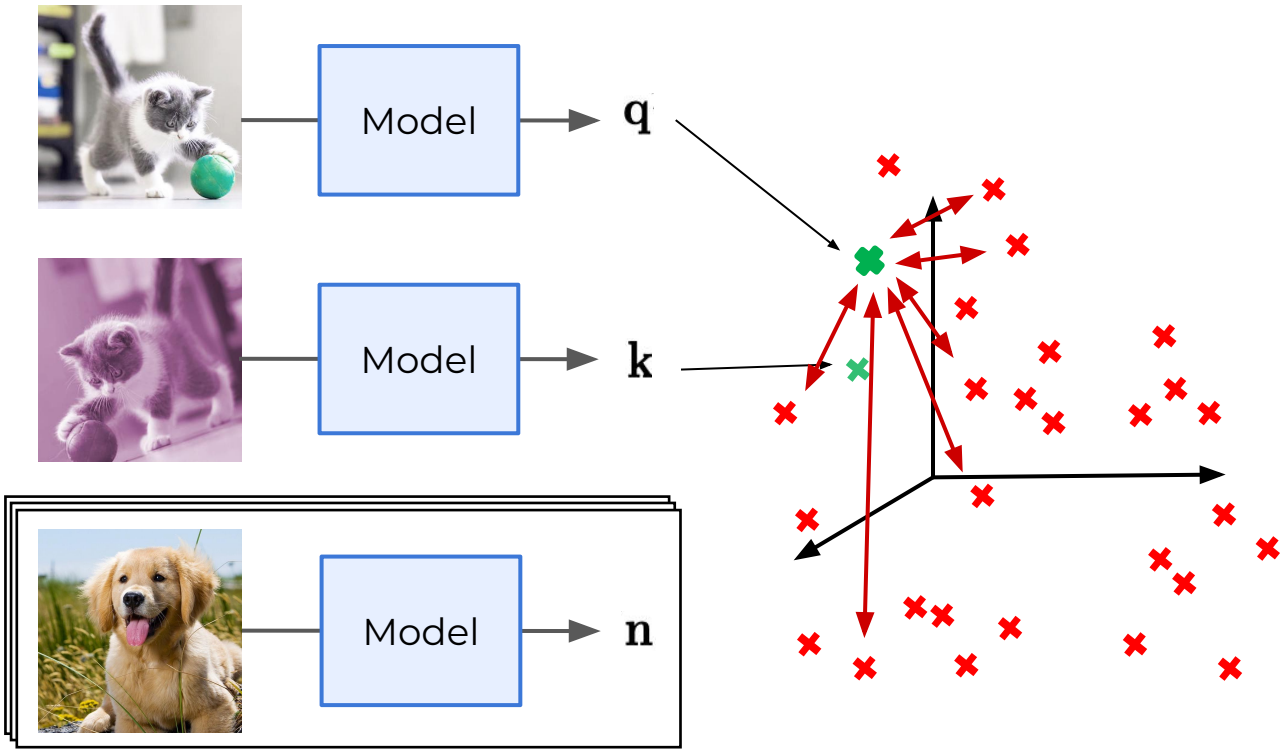
Contrastive self-supervised learning



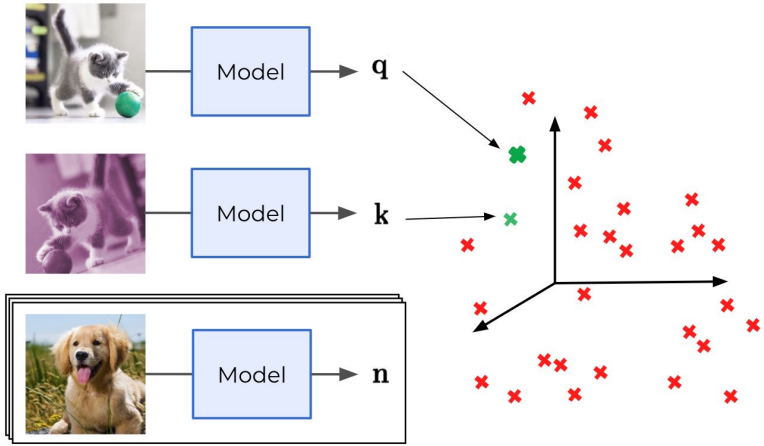
Contrastive self-supervised learning



Contrastive self-supervised learning



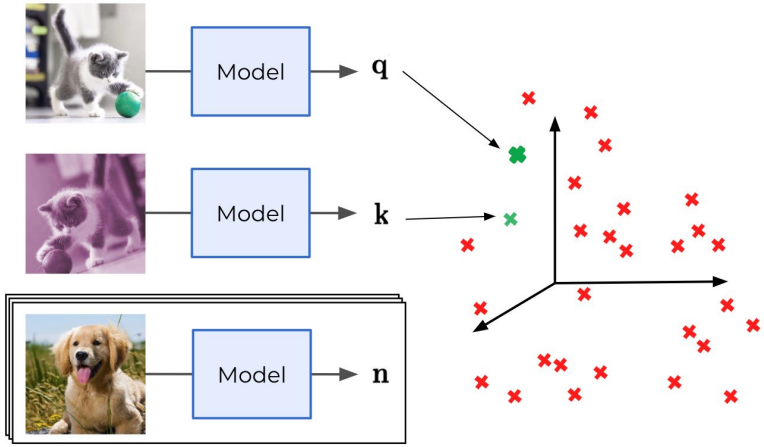
The InfoNCE loss



$$\mathcal{L}_{\mathbf{q},\mathbf{k},Q} = -\log \frac{\exp(\mathbf{q}^T\mathbf{k}/\tau)}{\exp(\mathbf{q}^T\mathbf{k}/\tau) + \sum_{\mathbf{n}\in Q}\exp(\mathbf{q}^T\mathbf{n}/\tau)},$$

[**CPC**] Oord, Aaron van den, et al. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[**MoCo**] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

Where do negatives come from?



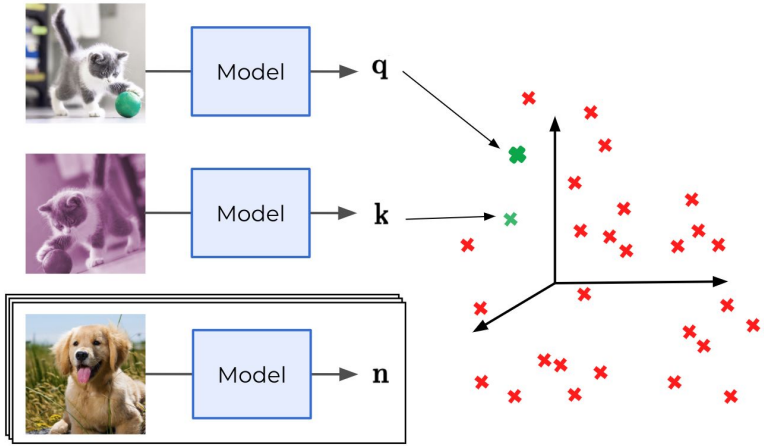
Negatives: Any other image [**Exemplar-CNN, InstDiscr**]

[**SimCLR**]: images from the same batch

[**MoCo**]: queue with images from last batches

[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[**MoCo-v2**] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[**Exemplar-CNN**] Dosovitskiy, et al. "Discriminative unsupervised feature learning with exemplar convolutional neural networks." TPAMI 2015
[**InstDiscr**] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.

A more challenging proxy task



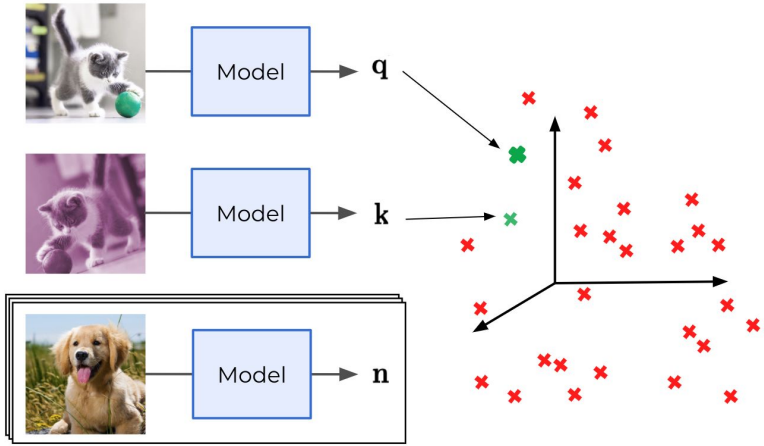
Key observation:

Making the proxy task more ***challenging*** leads to representations that generalize better

[**MoCo-v2**, **SimCLR**, **InfoMin Aug**]

[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[**MoCo-v2**] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[**InfoMin Aug.**] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

A more challenging proxy task



Key observation:

Making the proxy task more ***challenging*** leads to representations that generalize better

[**MoCo-v2**, **SimCLR**, **InfoMin Aug**]

How?

- More challenging positive pairs

[**SimCLR**]

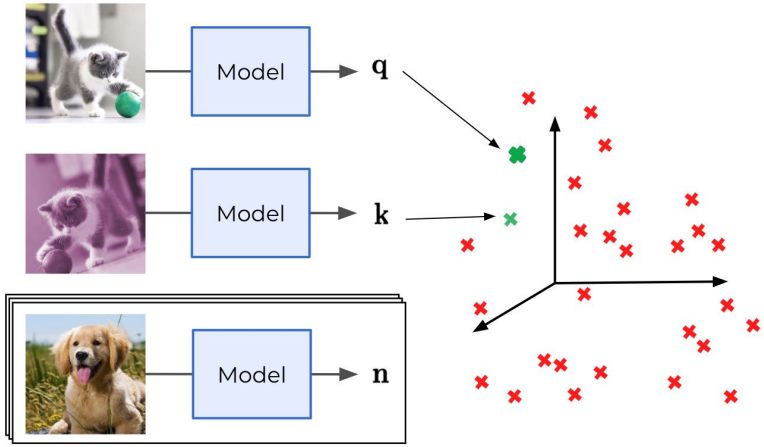
PyTorch-style data augmentation

```
RandomResizedCrop(scale=(0.2, 1.0))
RandomHorizontalFlip()
# CJ(x): random color jitter with x
cj = ColorJitter([0.8,0.8,0.8,0.4]*x)
RandomApply([cj], p=0.8)
# Blur: random blurring
blur = Blur(sigma=(0.1,2.0))
RandomApply([blur], p=0.5)
# RA: RandAugment
rnd_augment()
RandomGrayscale(p=0.2),
```

[**InfoMin Aug.**]

[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[**MoCo-v2**] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[**InfoMin Aug.**] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

A more challenging proxy task



Key observation:

Making the proxy task more ***challenging*** leads to representations that generalize better

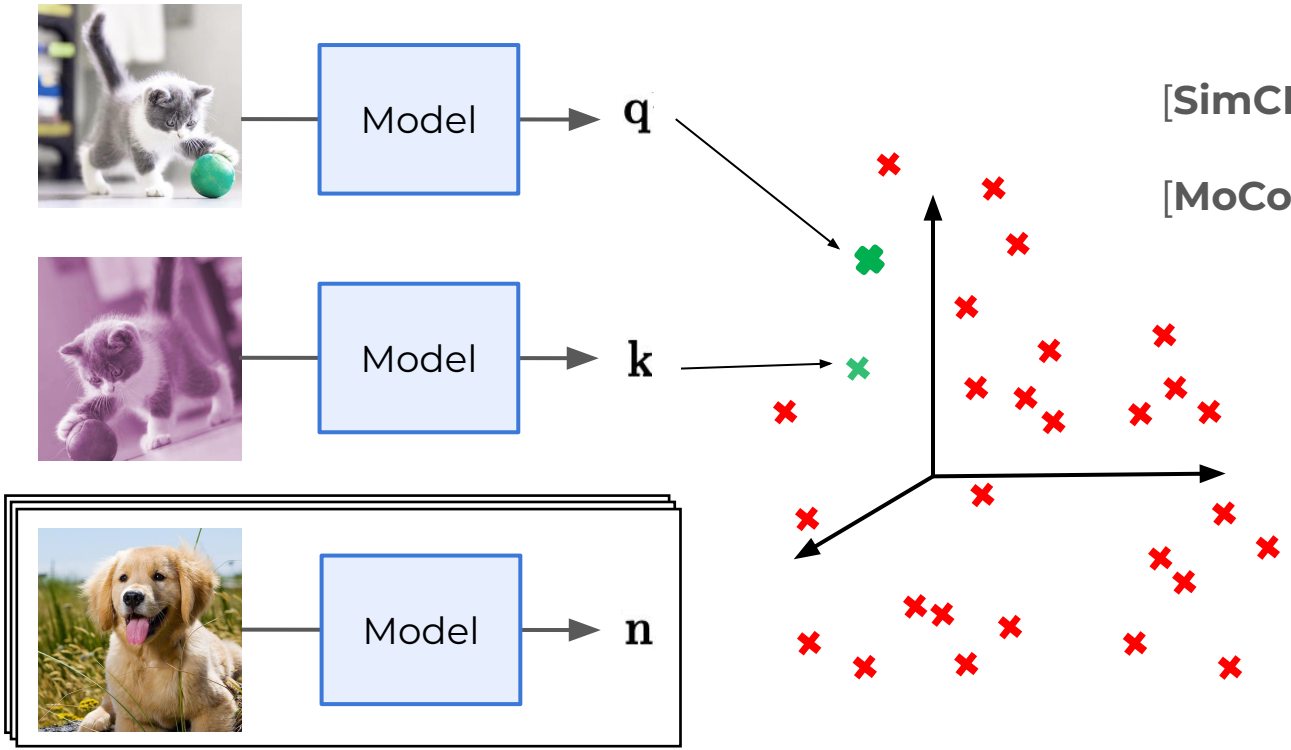
[**MoCo-v2**, **SimCLR**, **InfoMin Aug**]

How?

- More challenging positive pairs
- More challenging negatives

[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[**MoCo-v2**] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)
[**InfoMin Aug.**] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

How to get more challenging negatives?

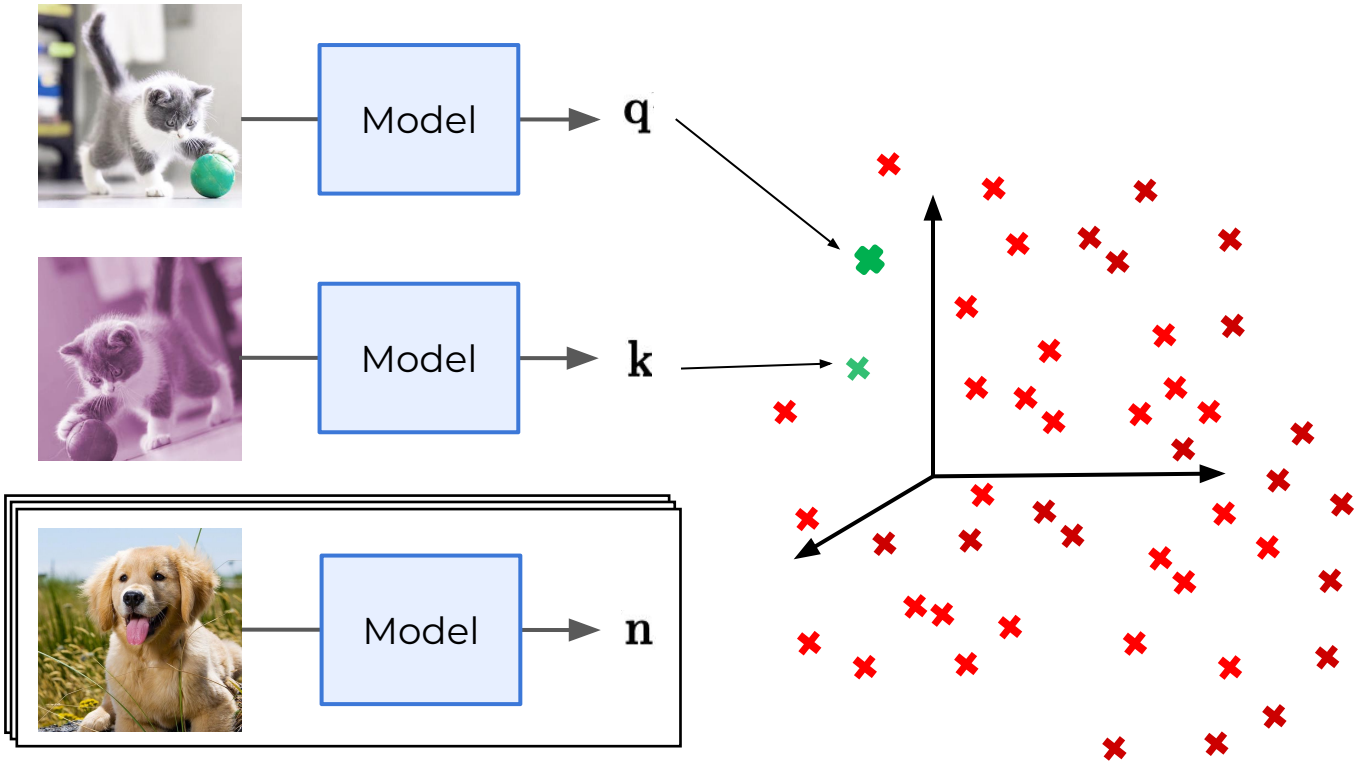


[**SimCLR**]: increase the batch size

[**MoCo**]: increase size of queue

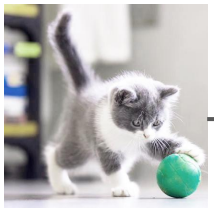
[**MoCo**] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.

How to get more challenging negatives?



[**MoCo**] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

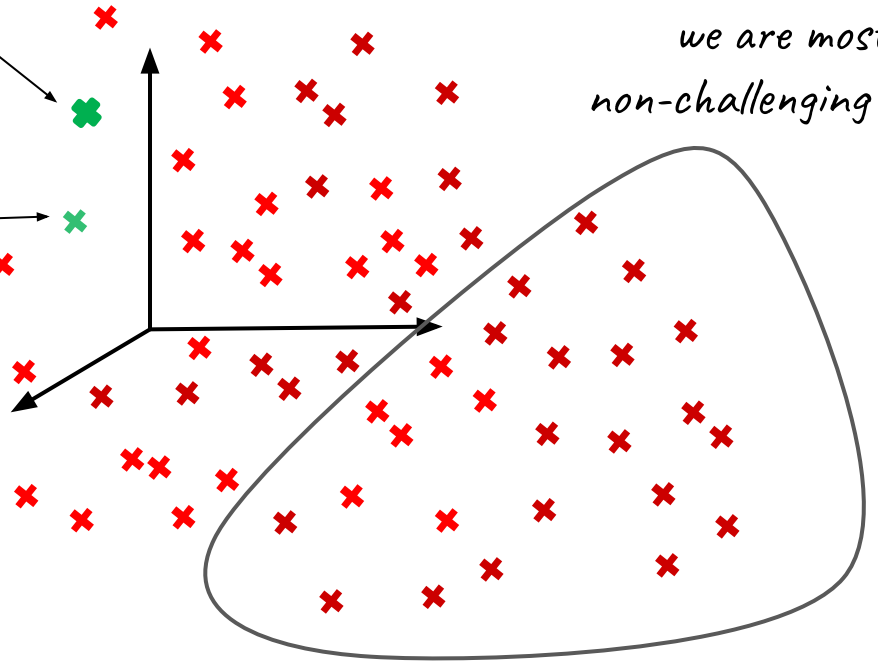
How to get more challenging negatives?



Model → $\mathbf{q}$

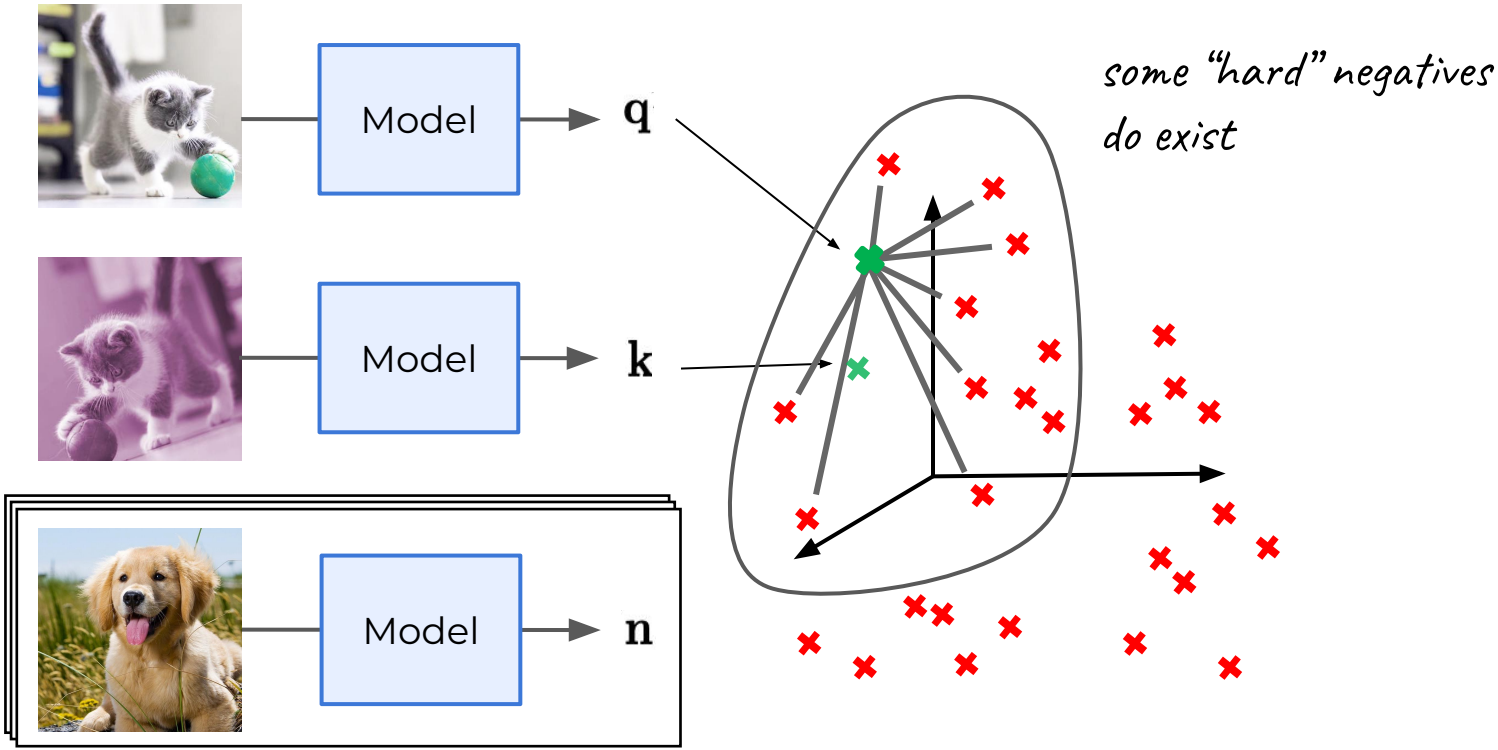
Model → $\mathbf{k}$

Model → $\mathbf{n}$

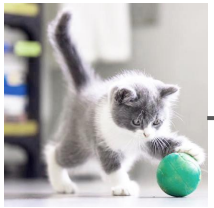


[**MoCo**] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.

Contrastive self-supervised learning



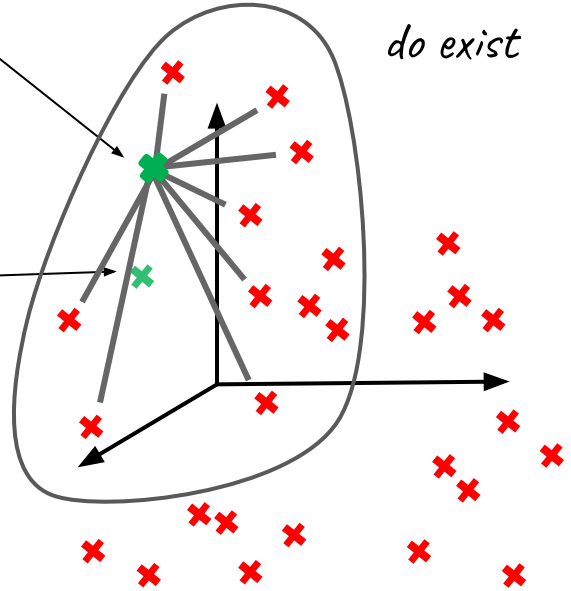
Contrastive self-supervised learning



Model → **q**

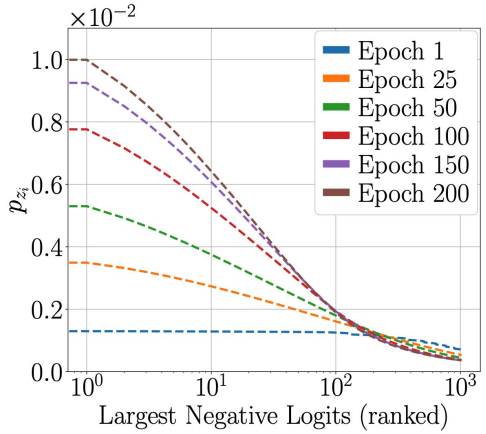
Model → **k**

Model → **n**



some "hard" negatives do exist

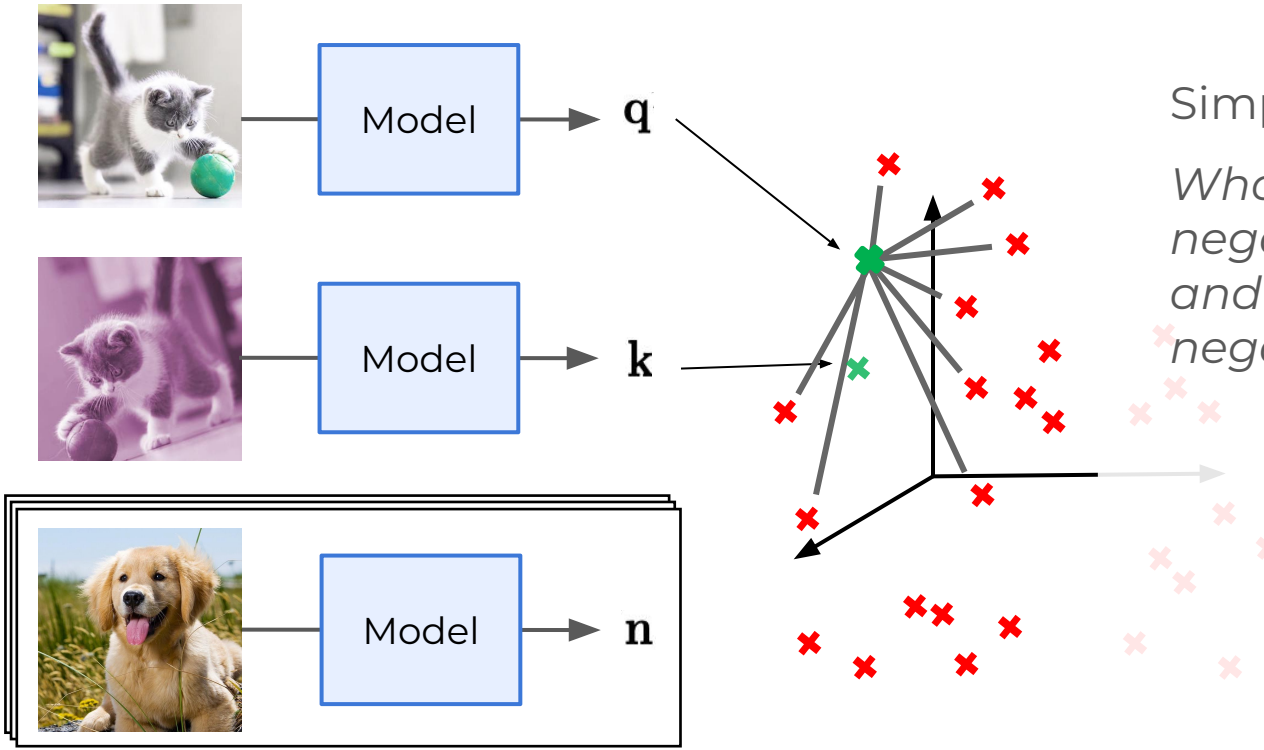
...and have strong influence



highest negative logits for [**MoCo-v2**] across epochs

[**MoCo-v2**] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv (2020)

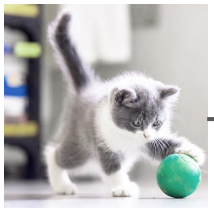
Mixing of Contrastive Hard Negatives



Simple idea:

What if we mix the hardest negatives for each query and synthesize new hard negatives?

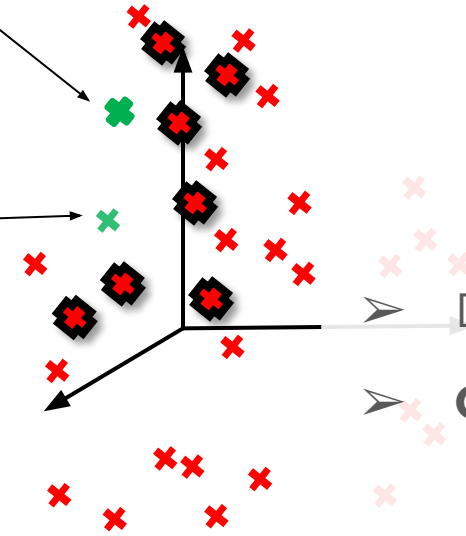
Mixing of Contrastive Hard Negatives



Model → $\mathbf{q}$

Model → $\mathbf{k}$

Model → $\mathbf{n}$

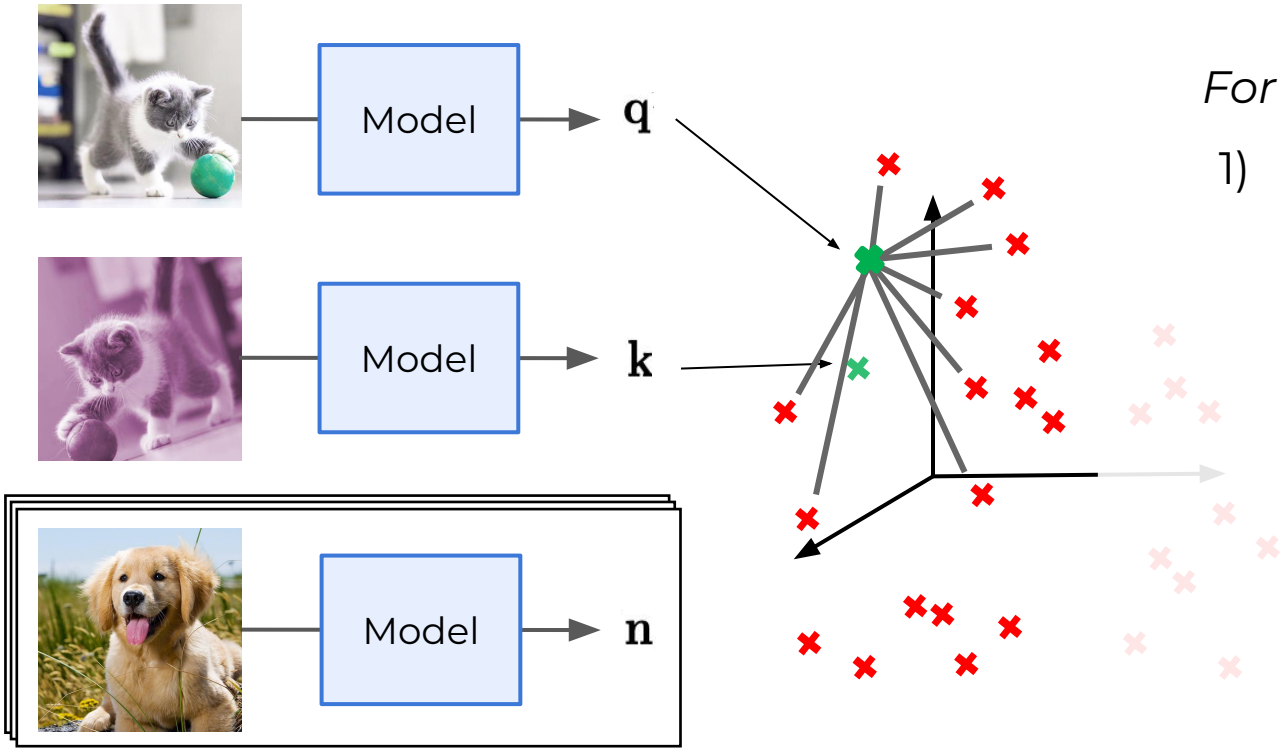


: synthetic hard negatives

MoCHi: mix the hardest negatives for each query and synthesize a small set of new (hard) negatives

- Directly **in feature space**
- **On-the-fly** for each query

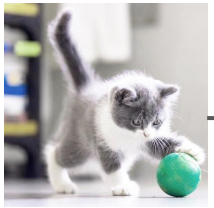
Mixing of Contrastive Hard Negatives



For each query **q**:

1) Randomly sample from closest negatives

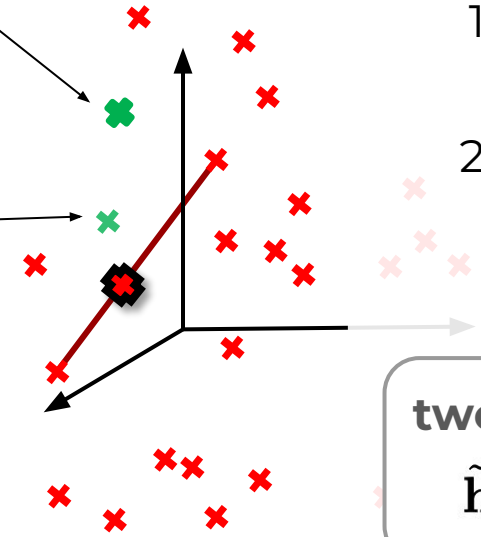
Mixing of Contrastive Hard Negatives



Model → $\mathbf{q}$

Model → $\mathbf{k}$

Model → $\mathbf{n}$



: synthetic hard negatives

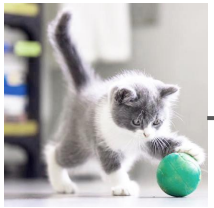
For each query **q**:

1) Randomly sample from closest negatives
2) synthesize a new negative by mixing …

two existing **negatives**:

$$\tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j$$

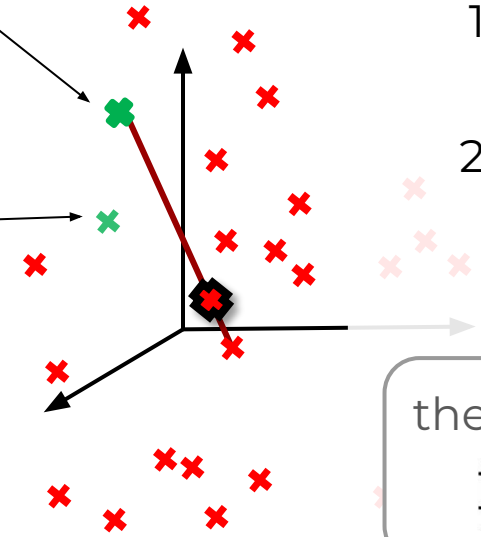
Mixing of Contrastive Hard Negatives



Model → $\mathbf{q}$

Model → $\mathbf{k}$

Model → $\mathbf{n}$



: synthetic hard negatives

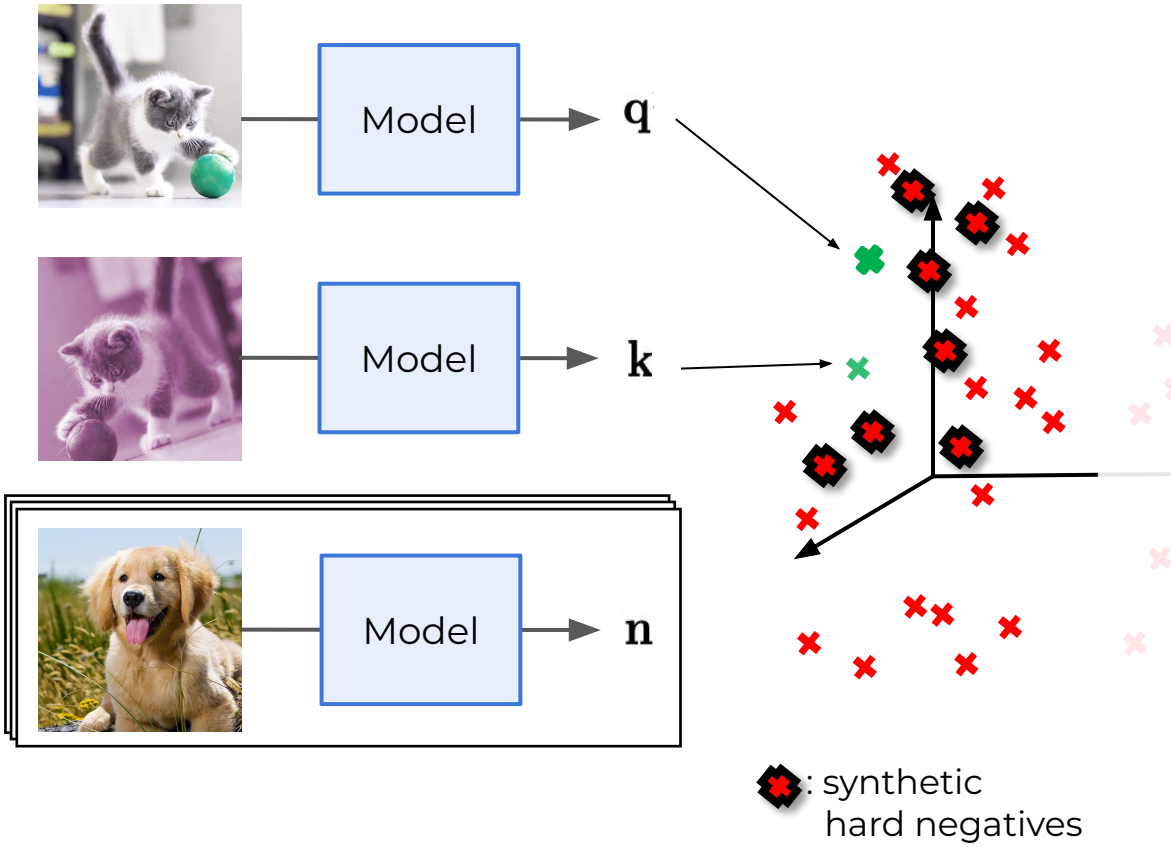
For each query **q**:

1) Randomly sample from closest negatives
2) synthesize a new negative by mixing …

the **query** with a **negative**:

$$\tilde{\mathbf{h}}'_k = \beta_k \mathbf{q} + (1-\beta_k)\mathbf{n}_j$$

Mixing of Contrastive Hard Negatives



For each query **q**:

1) Randomly sample from closest negatives
2) synthesize a new negative by mixing
3) Repeat!

Mixing of Contrastive Hard Negatives

...or MoCHi 

- We implement MoCHi on top of [**MoCo-v2**]

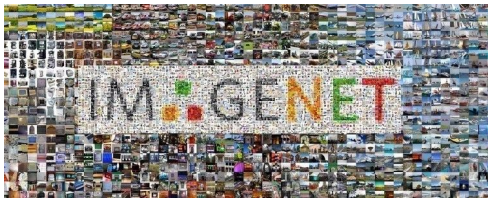
```
For every query q:

    # MoCo: calculate logits to the key and all negatives from the memory queue

    For s synthetic hard negatives:
        1: randomly sample 2 of the closest N negatives and a mixing coefficient
        2: “mix” and apply L2-normalization
        3: calculate logit (cosine similarity to q) append to the set of negative logits
```

- Small computational overhead
  s is orders of magnitude smaller than the memory queue size

[**MoCo-v2**] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv (2020)

Experimental evaluation

Self-Supervised learning
(Learn augmentation invariance)



images only (discard labels)

Model

Transfer Learning to
Downstream tasks

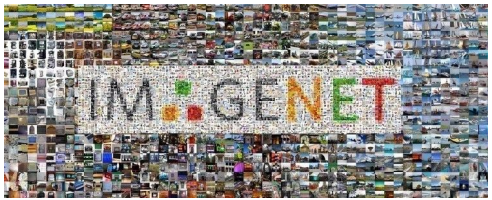
Image Classification (frozen)
ImageNet-1k or other datasets

Object Detection (fine-tune)
PASCAL VOC, MS COCO

Instance Segmentation (fine-tune)
MS COCO

Experimental evaluation

Self-Supervised learning
(Learn augmentation invariance)



images only (discard labels)

Model

Transfer Learning to
Downstream tasks

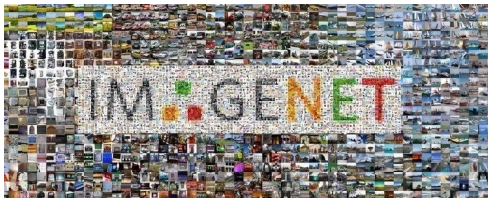
Image Classification
Datasets: ImageNet-100, ImageNet-1k

➢ MoCHi retains the strong performance of MoCo-v2 but shows little or no gains

Experimental evaluation

Transfer Learning to
Downstream tasks

Self-Supervised learning
(Learn augmentation invariance)



images only (discard labels)

Model

Object Detection &
Instance Segmentation

Datasets: PASCAL VOC, MS COCO

- MoCHi helps the model learn **faster**
- Strong performance after *only 100 epochs* of pre-training
- MoCHi after 200 epochs performs similar to MoCo-v2 after 800 epochs
- Gains persist after longer training (800 epochs)

Further analysis in our paper

- Analysis using a **class label "oracle"** (the ImageNet-1K labels)
  - How many synthetic points definitely come from the same class?
    - Only in very small percentage, not growing a lot during training
  - Using the labels: What if we discard negatives from the same class?
    - ImageNet-1K performance increases, approaches CE supervised case
- MoCHi results to **better "utilization"** of the embedding space
  - Measure the alignment and uniformity scores from [Wang & Isola]
    - MoCHi results to better utilization of the embedding space (uniformity)

Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., Larlus, D.
Hard negative mixing for contrastive learning
NeurIPS 2020

[**Wang & Isola**] Wang, Tongzhou, and Phillip Isola. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere." ICML 2020.

Take home message

- **Synthesize hard negatives** for a more challenging proxy task
- Faster self-supervised learning
- Performance gains for after fine-tuning over the baseline

https://europe.naverlabs.com/mochi

Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., Larlus, D.
Hard negative mixing for contrastive learning
NeurIPS 2020

Part 2:
Twin learning
for dimensionality reduction

NAVER LABS

Self-supervised learning beyond pixel inputs

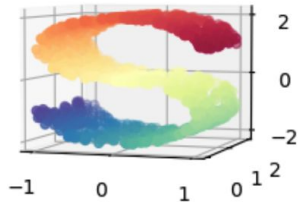
Representation learning → Dimensionality reduction

Assumption:

we have a (meaningful) input vector space we want to compress

Task:

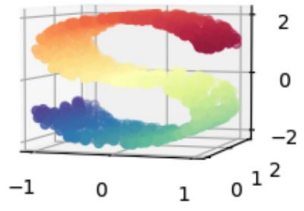
*Learn a **low-dimensional** vector space that preserves properties (e.g. topology) of a high-dimensional input vector space*



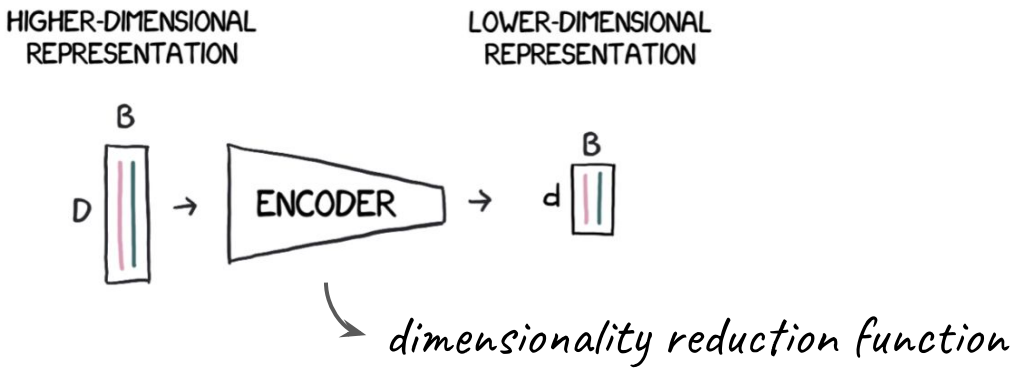
Dimensionality reduction

Why is this important?

- Compact representations are more memory and compute efficient
- Cannot afford to or don't want to fine-tune "end-to-end"
- Hard or impossible to hand-craft priors
- Dimensionality reduction is still used in practice in many fields
  - Computational biology and medicine
  - Remote sensing, climate and earth observation
  - Finance



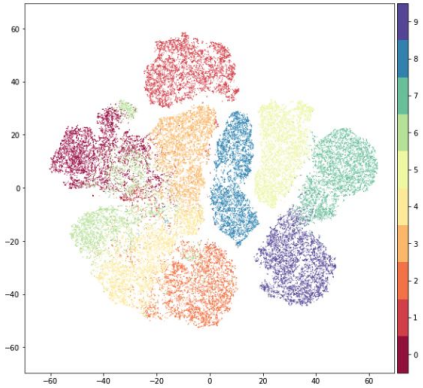
Dimensionality reduction



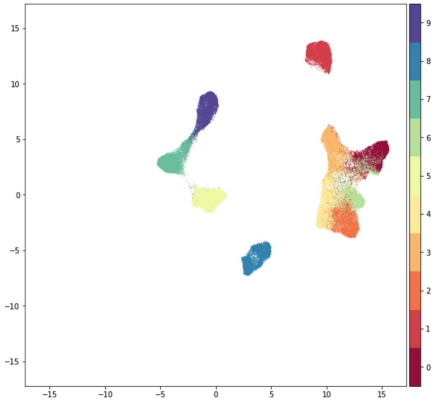
Dimensionality reduction for visualization

Visualization (output dimension of d = 2 or d = 3):

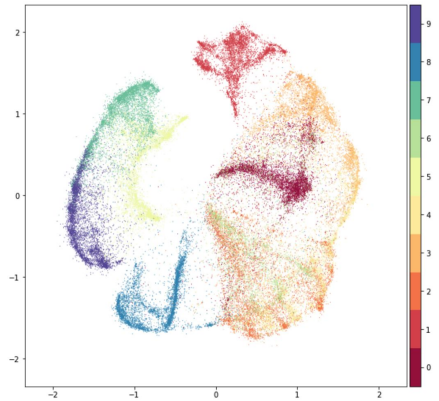
Many good methods specialize [**t-SNE**], [**UMAP**], [**MDE**]



t-SNE



UMAP



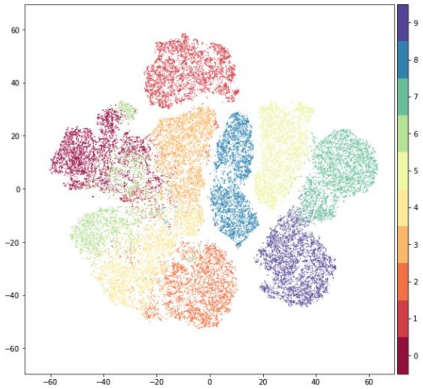
MDE

[**t-SNE**] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." JMLR 2008.
[**UMAP**] McInnes, *et al.* "UMAP: Uniform manifold approximation and projection for dimension reduction." arxiv 2018.
[**MDE**] A. Agrawal, A. Ali, and S. Boyd. Minimum-distortion embedding. arXiv preprint arXiv:2103.02559, 2021.

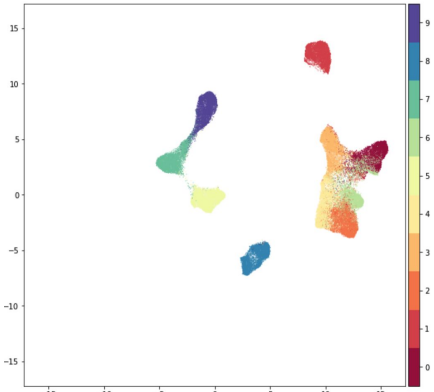
Dimensionality reduction for visualization

Visualization (output dimension of d = 2 or d = 3):

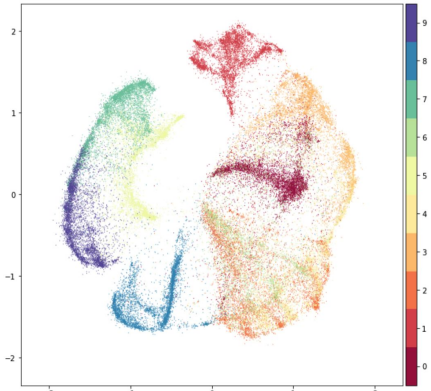
Many good methods specialize [**t-SNE**], [**UMAP**], [**MDE**]



t-SNE



UMAP



MDE

don’t scale well w.r.t. output dimension / made for visualization

[**t-SNE**] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." JMLR 2008.
[**UMAP**] McInnes, *et al.* "UMAP: Uniform manifold approximation and projection for dimension reduction." arxiv 2018.
[**MDE**] A. Agrawal, A. Ali, and S. Boyd. Minimum-distortion embedding. arXiv preprint arXiv:2103.02559, 2021.

For higher output dimensions (d > 3):

Manifold learning methods

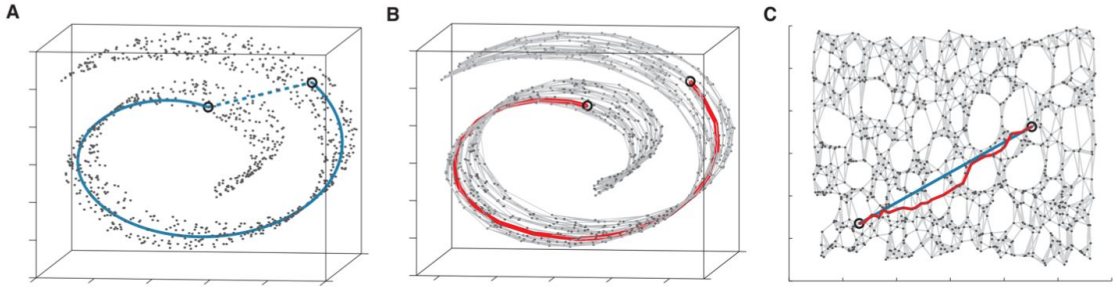


Figure from [**ISOMAP**]

[**ISOMAP**] Joshua B. Tenenbaum, Vin de Silva, John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science. 2000

[**scikit-learn**] has implementations and a nice overview of common manifold learning methods

For higher output dimensions (d > 3):

Manifold learning methods

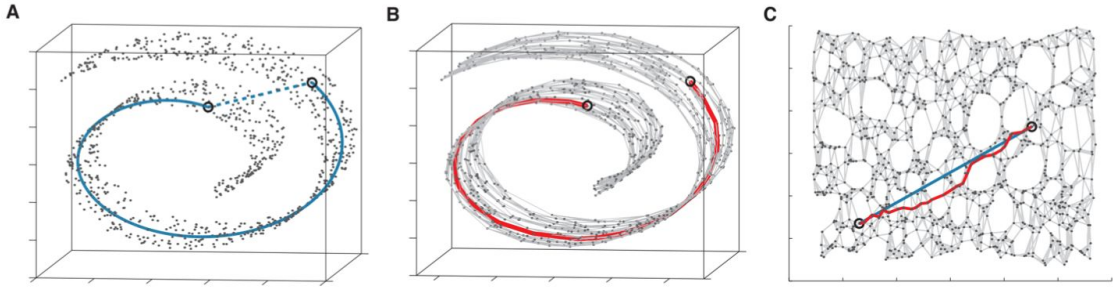


Figure from [**ISOMAP**]

don’t scale well w.r.t. dataset size

[**ISOMAP**] Joshua B. Tenenbaum, Vin de Silva, John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science. 2000

[**scikit-learn**] has implementations and a nice overview of common manifold learning methods

Dimensionality reduction

How about large-scale datasets?

Linear dimensionality reduction:

Principal component analysis [**PCA**]

- Is used in practice for large-scale systems
- Is actively used outside core-AI
  - biology (drug production, pollution detection, etc.),
  - remote sensing,
  - assisted medical diagnosis,
  - medical imaging analysis, etc.

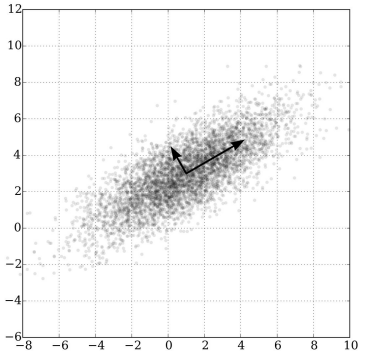


Figure from Wikipedia

[**PCA**] K. Pearson. “On lines and planes of closest fit to systems of points in space.” The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1901.

Why don’t manifold learning methods scale?

NAVER LABS

Manifold learning methods are **not scalable** (especially wrt dataset size):

- require propagation on k-NN *graphs* (many)
- use complex optimization solvers (many)
- eigen-decompositions (many)

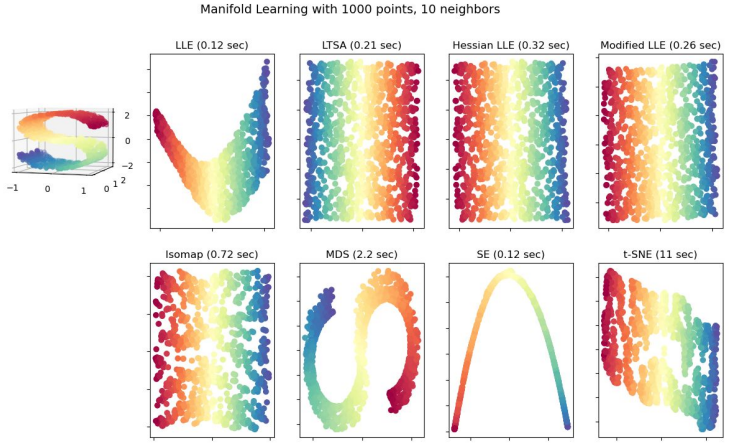


Figure from [**scikit-learn**]

[**scikit-learn**] has implementations and a nice overview of common manifold learning methods

Why don’t manifold learning methods scale?

Manifold learning methods are **not scalable**

(especially wrt dataset size):

- require propagation on k-NN *graphs* (many)
- use complex optimization solvers (many)
- eigen-decompositions (many)

Not a issue for recent self-supervised visual representation learning frameworks

(SGD solvers,
simple contrastive losses)

Why don’t manifold learning methods scale?

Manifold learning methods are **not scalable**

(especially wrt dataset size):

- require propagation on k-NN *graphs* (many)
- use complex optimization solvers (many)
- eigen-decompositions (many)

Not a issue for recent self-supervised visual representation learning frameworks

(SGD solvers,
simple contrastive losses)

Can we borrow from them to design dimensionality reduction approaches?

A suitable loss for dimensionality reduction

The [**Barlow Twins**] loss:

- **Simple** and **scalable**
- No contrasting pairs, only positives
- A loss function that fits well:
  - Decorrelation-focused
  - Trivially avoids collapsing
    (despite only using positive pairs)

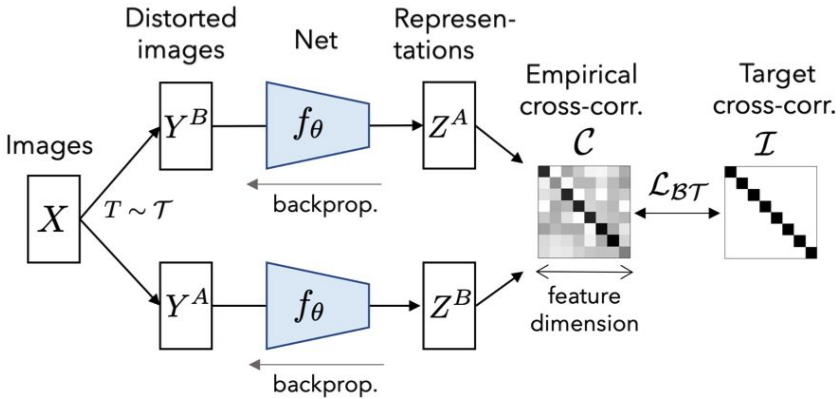
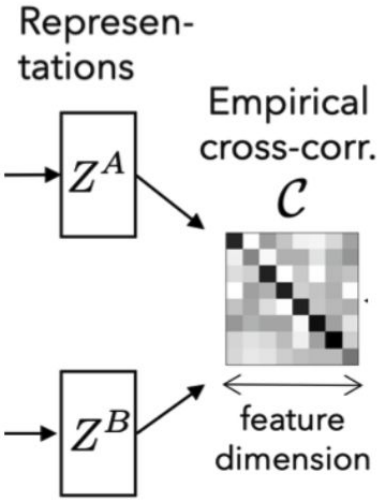


Figure from [**Barlow Twins**]

[**Barlow Twins**] Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction” ICML 2021.

The Barlow Twins loss



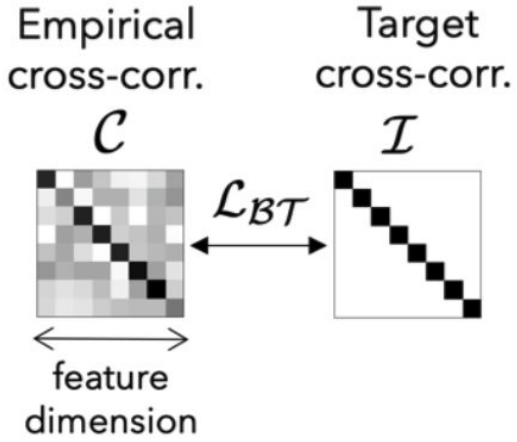
$$\mathcal{C}_{ij} = \frac{\sum_b \hat{z}^A_{b,i} \hat{z}^B_{b,j}}{\sqrt{\sum_b (\hat{z}^A_{b,i})^2} \sqrt{\sum_b (\hat{z}^B_{b,j})^2}}$$

Empirical (batch) cross-corr. matrix $\mathcal{C}$

- across the **feature dimension**
- outer product of the normalized representation for every positive pair
- averaged over the batch

[**Barlow Twins**] Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction” ICML 2021.

The Barlow Twins loss

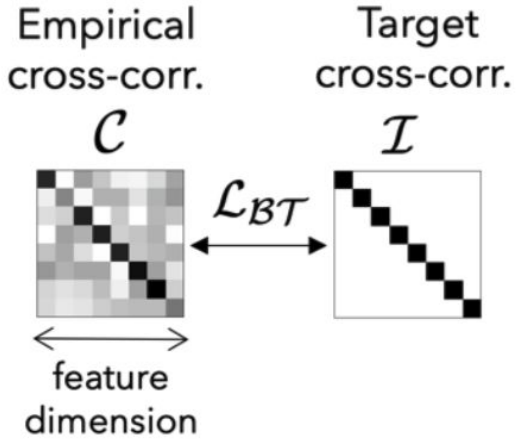


$$\mathcal{C}_{ij} = \frac{\sum_b \hat{z}^A_{b,i} \hat{z}^B_{b,j}}{\sqrt{\sum_b (\hat{z}^A_{b,i})^2} \sqrt{\sum_b (\hat{z}^B_{b,j})^2}}$$

$$\mathcal{L}_{BT} = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{push diagonal elements to 1}} + \lambda \underbrace{\sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2}_{\text{push off-diagonal elements to 0}}$$

[**Barlow Twins**] Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction” ICML 2021.

The Barlow Twins loss



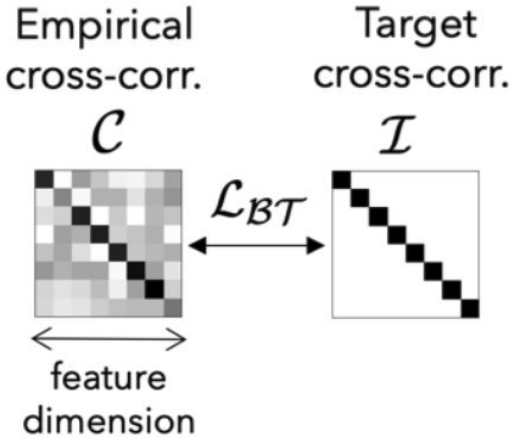
$$\mathcal{C}_{ij} = \frac{\sum_b \hat{z}^A_{b,i} \hat{z}^B_{b,j}}{\sqrt{\sum_b (\hat{z}^A_{b,i})^2} \sqrt{\sum_b (\hat{z}^B_{b,j})^2}}$$

$$\mathcal{L}_{BT} = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{i} + \lambda \sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2$$

maximize the dot product
of every positive pair

[**Barlow Twins**] Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction” ICML 2021.

The Barlow Twins loss



$$\mathcal{C}_{ij} = \frac{\sum_b \hat{z}^A_{b,i} \hat{z}^B_{b,j}}{\sqrt{\sum_b (\hat{z}^A_{b,i})^2} \sqrt{\sum_b (\hat{z}^B_{b,j})^2}}$$

$$\mathcal{L}_{BT} = \sum_i (1 - \mathcal{C}_{ii})^2 + \lambda \underbrace{\sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2}$$

de-correlate output dimensions

[**Barlow Twins**] Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction” ICML 2021.

(Our only) Assumption:

We have a “meaningful” input space that we want to compress

Pre-text task for generic input spaces:

(used to define positive pairs)

- Add noise (e.g. denoising autoencoders [**DAE**])
- Neighborhood Embedding: nearest neighbors [**DrLim, t-SNE ++**]

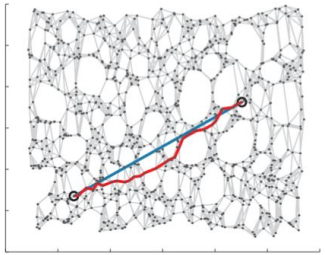
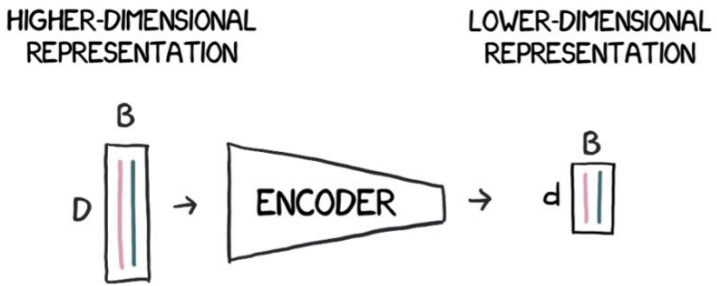
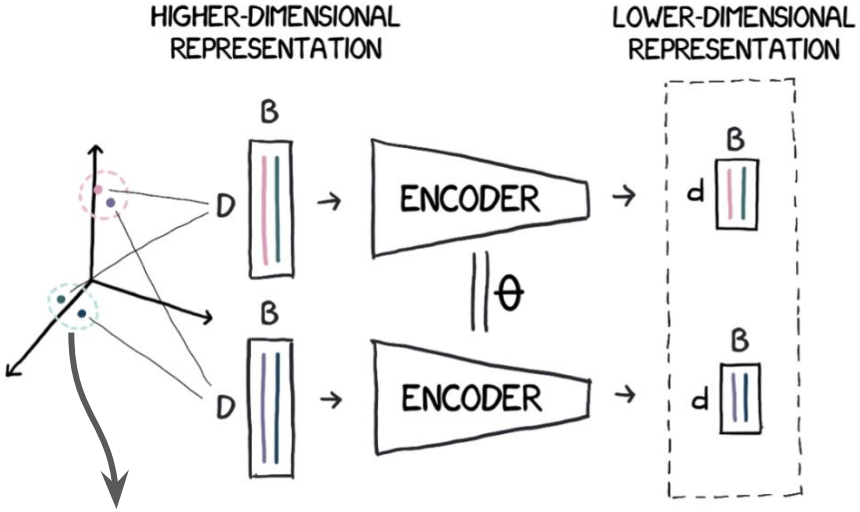


Figure from [**Isomap**]

[**DAE**] P Vincent *et al.* "Extracting and composing robust features with denoising autoencoders." ICML 2008.
[**Isomap**] Joshua B. Tenenbaum, Vin de Silva, John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science. 2000
[**t-SNE**] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." JMLR 2008.
[**DrLim**] Hadsell, *et al.* “Dimensionality reduction by learning an invariant mapping.” CVPR 2006.

Towards scalable dimensionality reduction

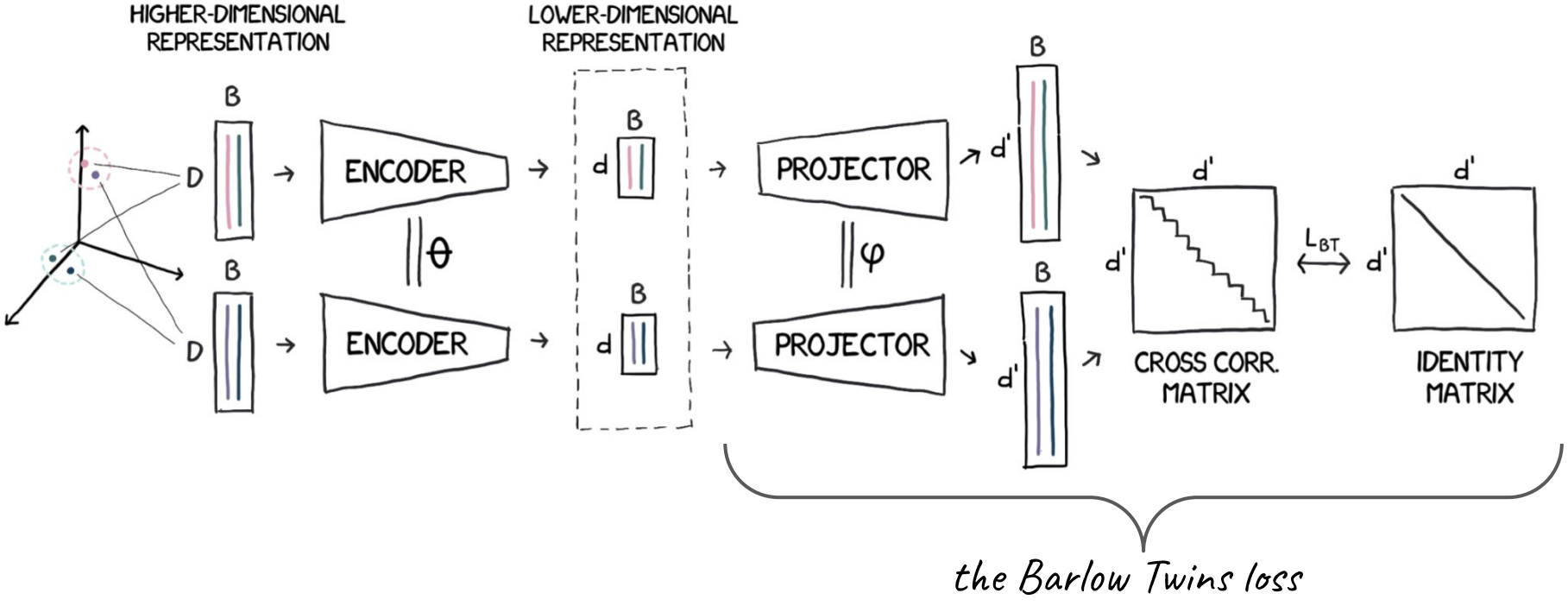




a positive pair
via nearest neighbors

Twin **L**earning for **D**imensionality **R**eduction

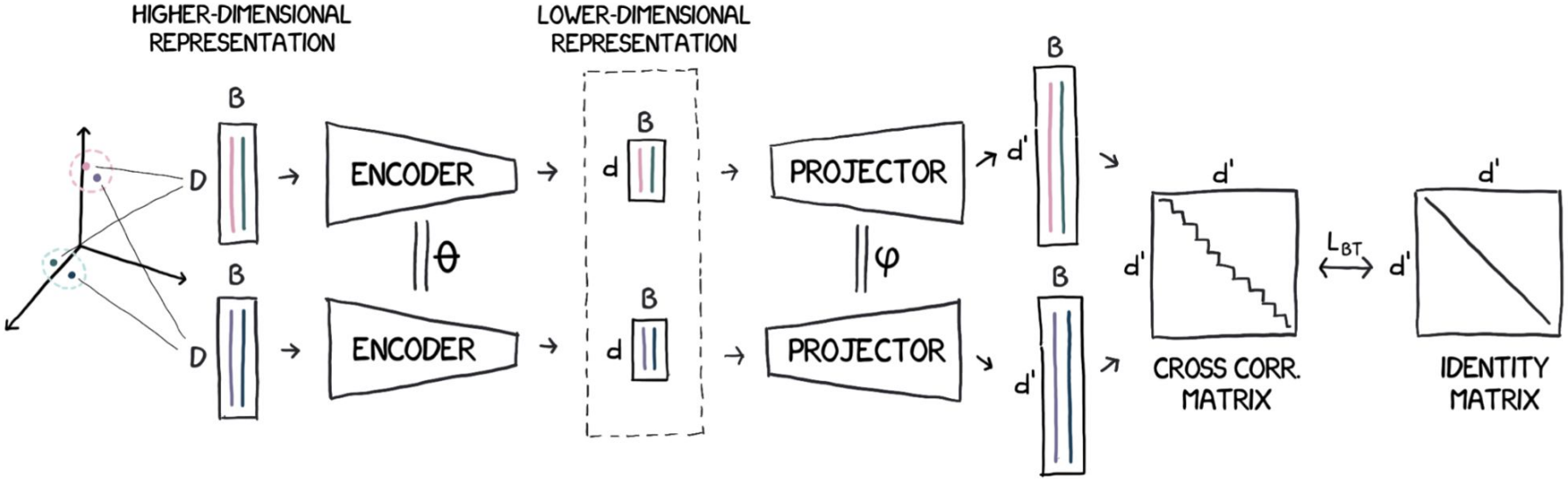
NAVER LABS



Twin **L**earning for **D**imensionality **R**eduction

NAVER LABS

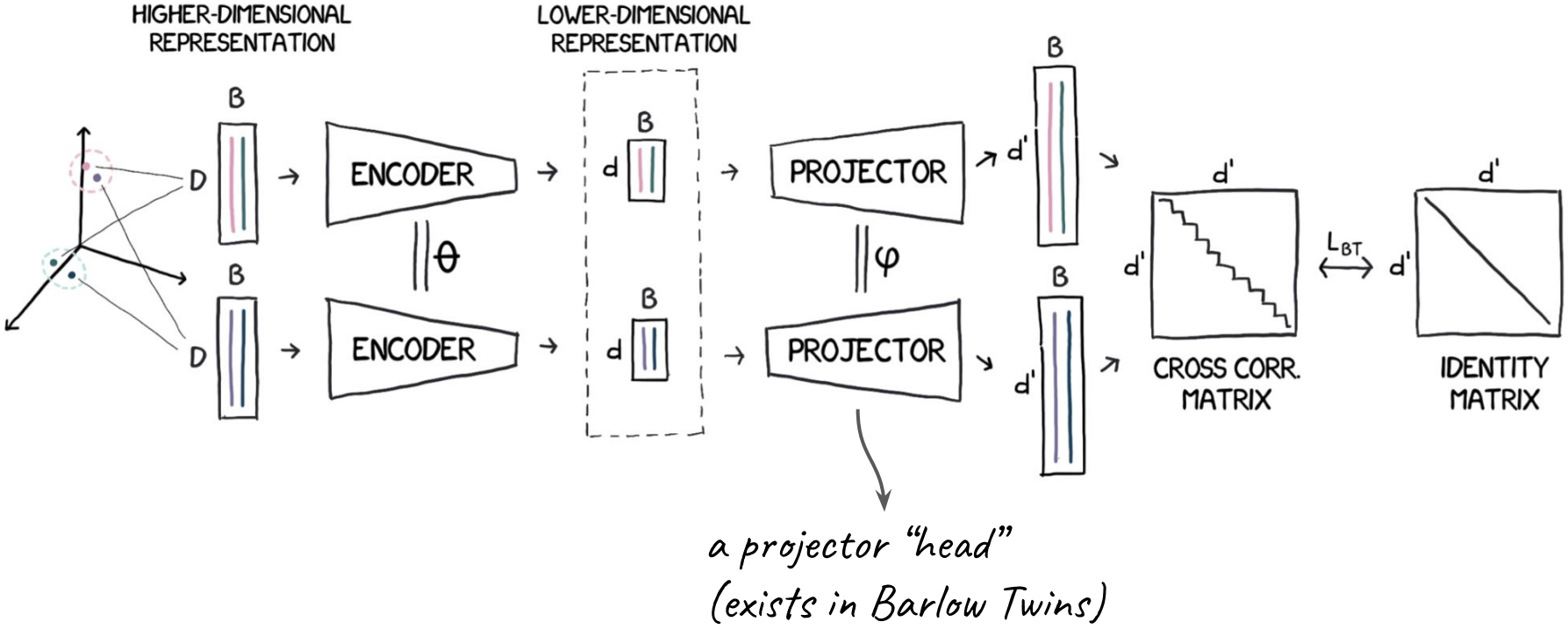
...or **TLDR**



Twin **L**earning for **D**imensionality **R**eduction

NAVER LABS

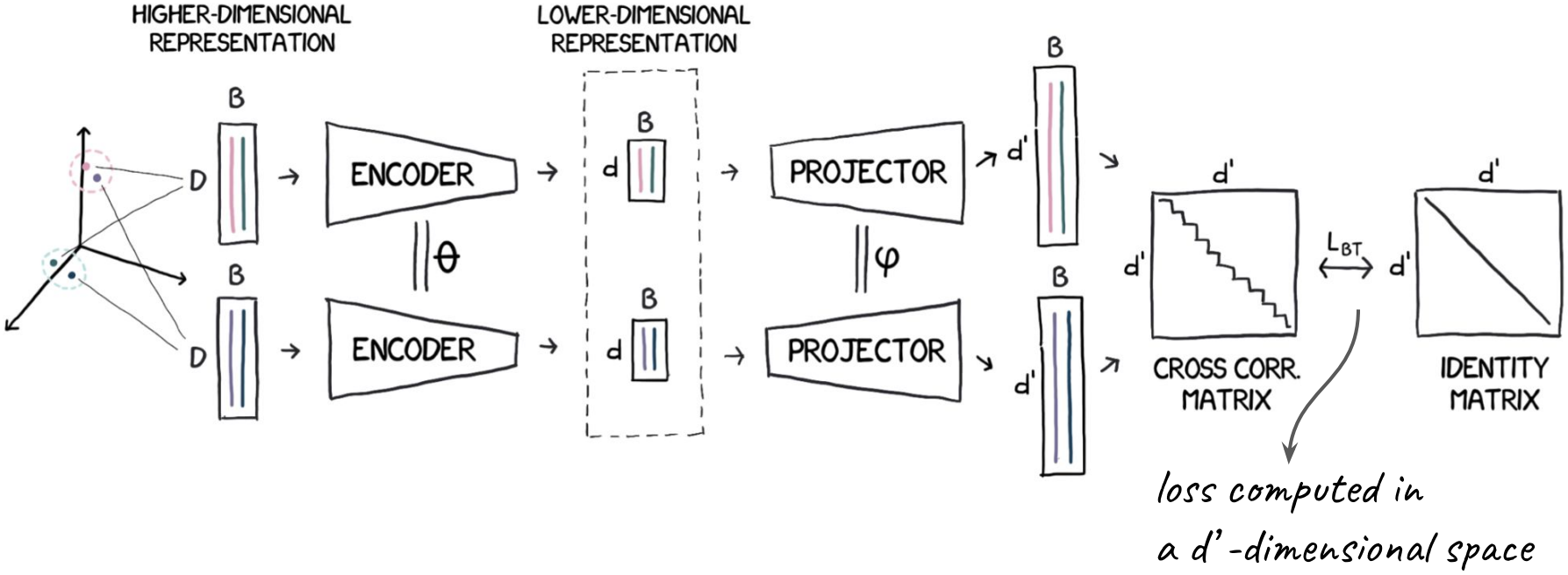
...or **TLDR**



Twin **L**earning for **D**imensionality **R**eduction

NAVER LABS

...or **TLDR**

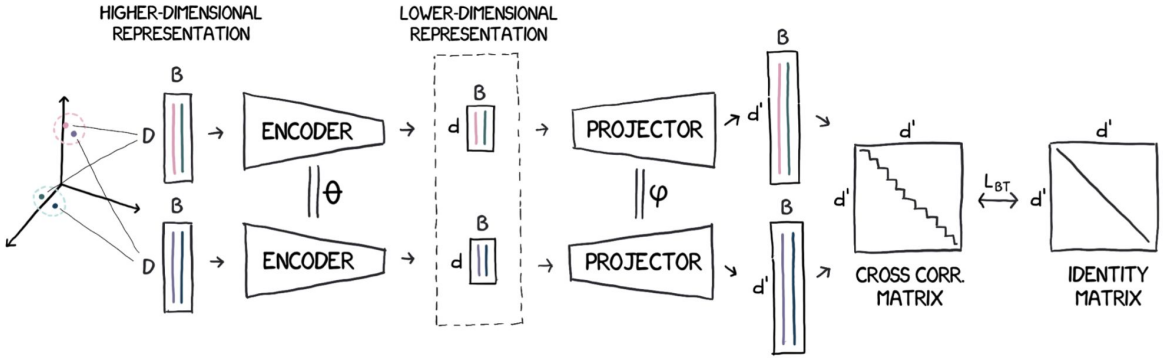


TLDR: Simple Algorithm

1: for every point x, calculate the **k**-nearest neighbors

2: Create positive pairs (x, y) by sampling y from the set of neighbors of x

3: Learn the parameters θ and φ by optimizing the [**Barlow Twins**] loss



Evaluation tasks & architectures choices

Large-scale retrieval

- Landmark image retrieval
- Document retrieval

Output dimensions:

$$64 \leq d \leq 256$$

Baseline method (used in practice):

PCA*: PCA + whitening

Train on large datasets

Architecture choices:

- **Linear encoder**
- 2-layer MLP projector with d' >> d

a query from the **RParis** dataset
and its "**medium**" and "**hard**" positives
(Figure adapted from [Radenović 2018])

Figure from Radenović, Filip, et al. "Revisiting oxford and paris: Large-scale image retrieval benchmarking." CVPR 2018.

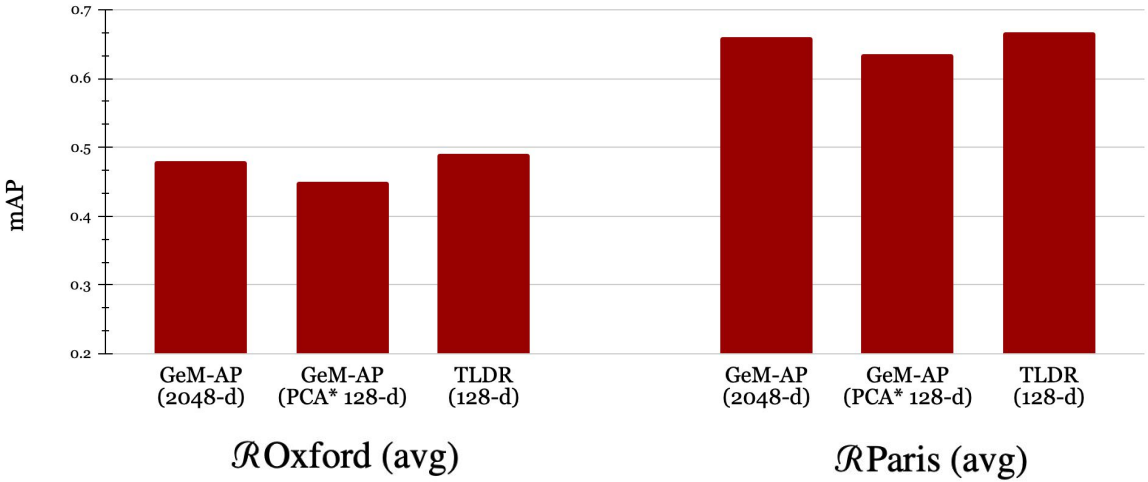
Results - Landmark image retrieval

- PCA* (PCA+whitening) is a part of state-of-the-art pipelines (eg [**GeM-AP**, **HOW**])
- We simply replace PCA* with **linear TLDR** (linear encoder)

TLDR (ROxford, 128-dim):

➢ + 4% mAP [**GeM-AP**]

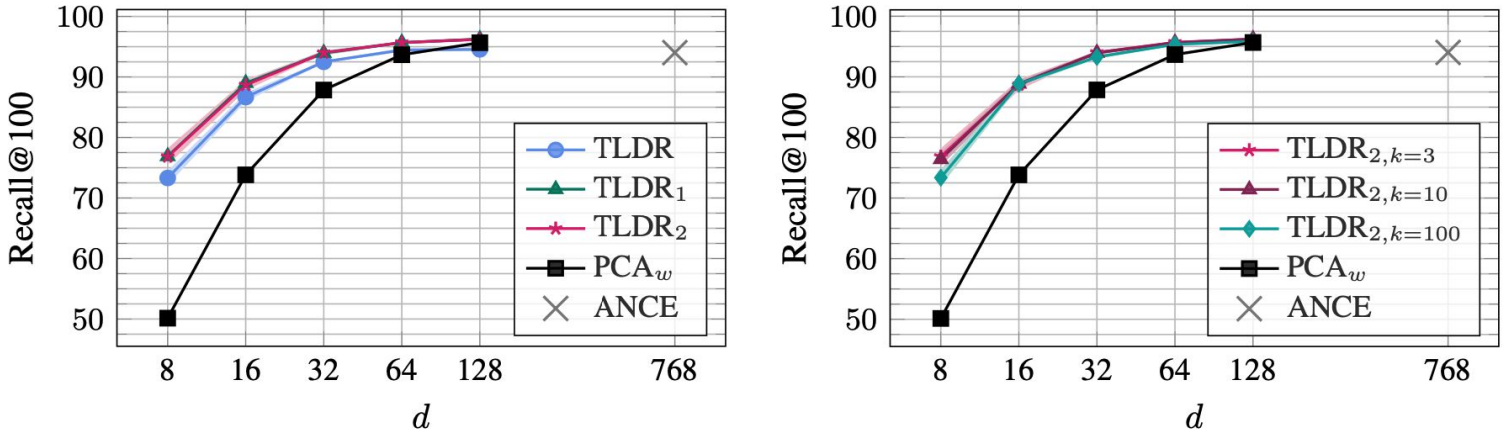
➢ match [**GeM-AP**] with 16 times fewer dims



[**GeM-AP**] J. Revaud et al. ”Learning with average precision: Training image retrieval with a listwise loss.” ICCV, 2019.
[**HOW**] Tolias et al. "Learning and aggregating deep local descriptors for instance-level recognition." ECCV 2020.

Results - Document retrieval

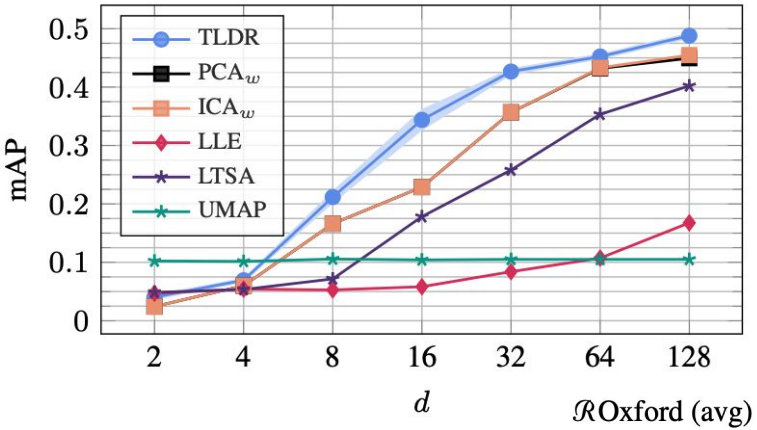
- Argument retrieval results on [**ArguAna**]
- Features from [**ANCE**]
  - Match performance of full representation using 4% of the dimensions (PCA needs double)



[**ANCE**] L. Xiong et al, "Approximate nearest neighbor negative contrastive learning for dense text retrieval", ICLR, 2021.
[**ArguAna**] Wachsmuth et al, "Retrieval of the best counterargument without prior topic knowledge", ACL 2018

Compared to manifold methods:

- **TDLR** outperforms all for d > 4
- Note: Linear dimensionality reduction is a very strong baseline for d > 8



Advantages of **TLDR**

Generality

- Can compress any meaningful vector input space (no further assumptions)
- many types of encoders (linear/factorized-linear/MLP)

Simplicity

- Robustness to hyper-parameters (also: parameters transfer across tasks)
- Easy-to-optimize loss [Barlow Twins] (no negatives, LARS optimizer)

Scalability

- Learning via mini-batch stochastic gradient descent (very GPU-friendly)
- High resilience to approximate nearest neighbors (+ no graph propagation)

TLDR: Easy-to-use code!

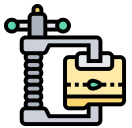
Public code with `scikit-learn` style API:

https://github.com/naver/tldr

```
from tldr import TLDR

tldr = TLDR(n_components=32, n_neighbors=10,
            encoder='linear', projector='mlp-1-2048', device='cuda')

tldr.fit(X, epochs=100, batch_size=1024)

Z = tldr.transform(X, l2_norm=True)
```

Y. Kalantidis, C. Lassance, J. Almazan, D. Larlus. **“TLDR: Twin Learning for Dimensionality Reduction”**. arXiv, 2021

Take home message



- **TLDR**: A scalable dimensionality reduction method
- Strong performance on many retrieval tasks
- Easy-to-use code publicly available
- *What is TLDR suitable for?*

  (*Linear)* dimensionality reduction to 32 - 256 dims
- *What is TLDR not suitable for?*

  Visualizations, representation learning (e.g. vs hand-crafting priors)

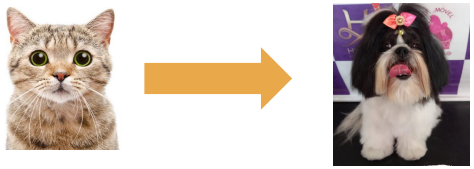
Y. Kalantidis, C. Lassance, J. Almazan, D. Larlus.
"TLDR: Twin Learning for Dimensionality Reduction"
arXiv, 2021

Part 3:
Measuring
Concept Generalization

NAVER LABS

Concept Generalization:

*The extent to which models trained on a set of (**seen**) visual concepts can be used to recognize a set of **unseen** target concepts*

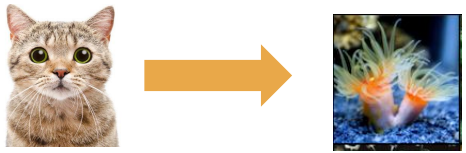


*Is a model that recognizes **cats** a good starting point for learning to recognize **dogs**?*

Dog model: Dr. Beluda

Concept Generalization:

*The extent to which models trained on a set of (**seen**) visual concepts can be used to recognize a set of **unseen** target concepts*



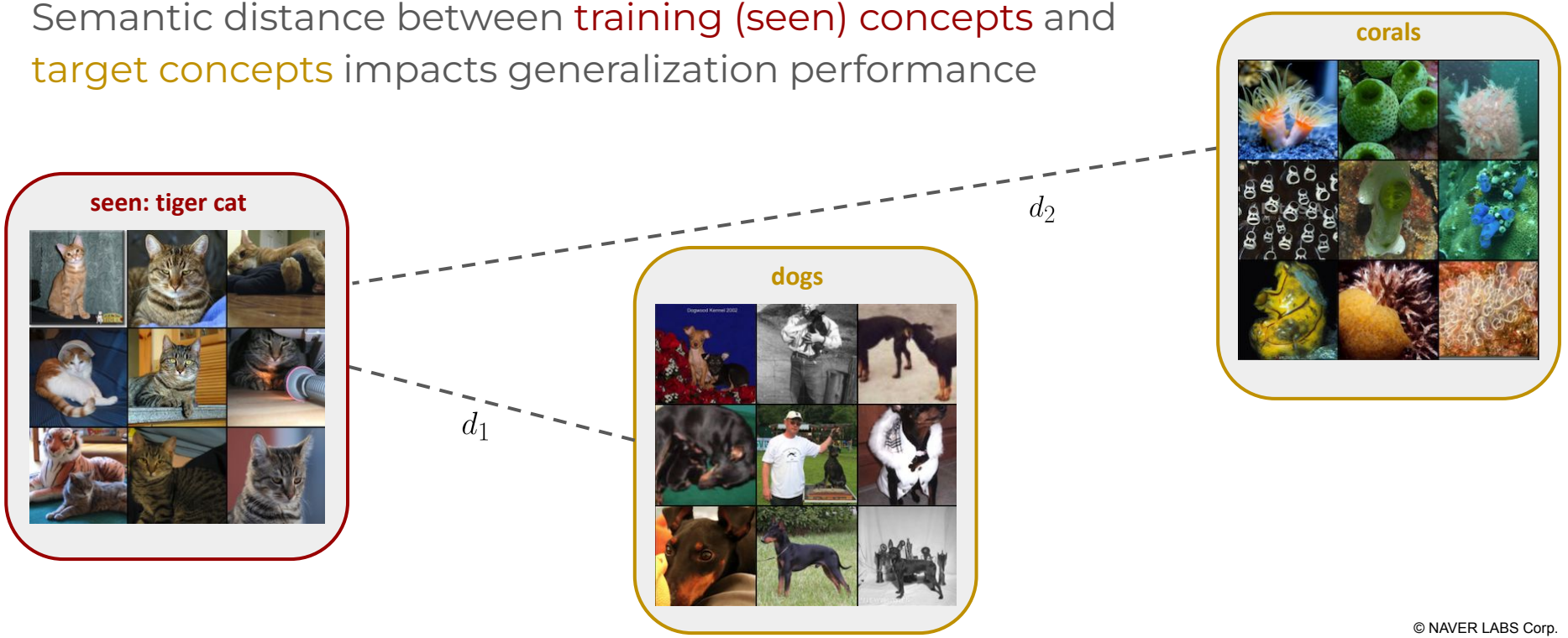
*Is a model that recognizes **cats** a good starting point for learning to recognize **corals**?*

Dog model: Dr. Beluda

The importance of “semantic distance”

Hypothesis:

Semantic distance between training (seen) concepts and target concepts impacts generalization performance



The importance of “semantic distance”

Hypothesis:

Semantic distance between training (seen) concepts and target concepts impacts generalization performance

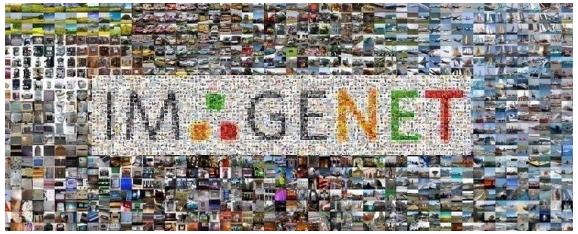
Goal:

Design a better **benchmark** for measuring concept generalization

How do we usually measure concept generalization?

Learn model parameters on ImageNet-1K

Model



Measure performance on (many) other datasets

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
BYOL (ours)	**75.3**	91.3	**78.4**	**57.2**	**62.2**	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR (repro)	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6
SimCLR [8]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised-IN [8]	72.3	**93.6**	78.3	53.7	61.9	66.7	**61.0**	**82.8**	74.9	**91.5**	**94.5**	94.7

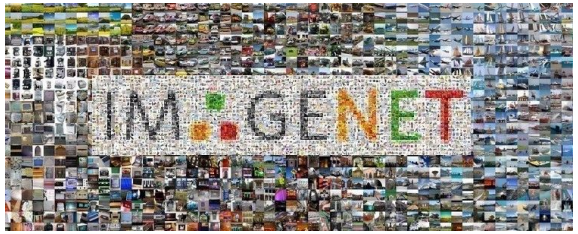
figure from [BYOL]

[BYOL] Grill, et al. "**Bootstrap your own latent: A new approach to self-supervised learning.**" NeurIPS, 2020.

How do we usually measure concept generalization?

Learn model parameters on ImageNet-1K

Model



Measure performance on (many) other datasets

Unclear how the concepts in these datasets relate to the concepts seen during training

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
BYOL (ours)	**75.3**	91.3	**78.4**	**57.2**	**62.2**	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR (repro)	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6
SimCLR [8]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised-IN [8]	72.3	**93.6**	78.3	53.7	61.9	66.7	**61.0**	**82.8**	74.9	**91.5**	**94.5**	94.7

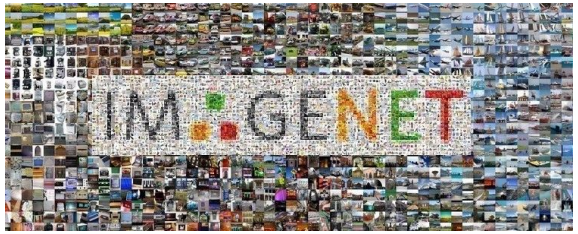
figure from [BYOL]

[BYOL] Grill, et al. "**Bootstrap your own latent: A new approach to self-supervised learning.**" NeurIPS, 2020.

How do we usually measure concept generalization?

Learn model parameters on ImageNet-1K

Model



Measure performance on (many) other datasets

Method	Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
BYOL (ours)	**75.3**	91.3	**78.4**	**57.2**	**62.2**	**67.8**	60.6	82.5	75.5	90.4	94.2	**96.1**
SimCLR (repro)	72.8	90.5	74.4	42.4	60.6	49.3	49.8	81.4	**75.7**	84.6	89.3	92.6
SimCLR [8]	68.4	90.6	71.6	37.4	58.8	50.3	50.3	80.5	74.5	83.6	90.3	91.2
Supervised-IN [8]	72.3	**93.6**	78.3	53.7	61.9	66.7	**61.0**	**82.8**	74.9	**91.5**	**94.5**	94.7

figure from [BYOL]

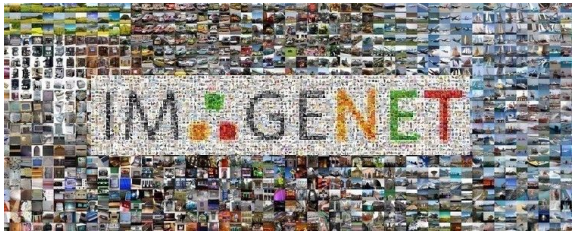
[BYOL] Grill, et al. "**Bootstrap your own latent: A new approach to self-supervised learning.**" NeurIPS, 2020.

How do we usually measure concept generalization?

Training or **seen** concepts

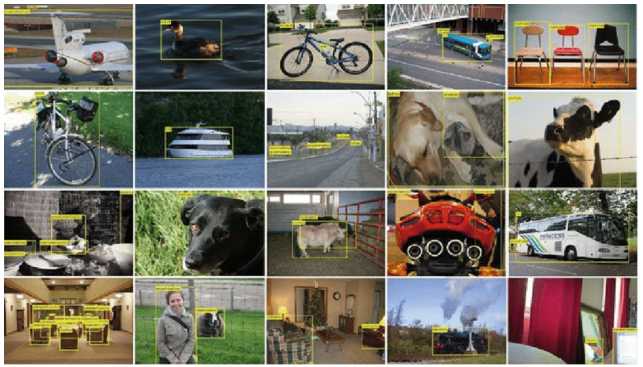
Learn model parameters on ImageNet-1K

Model



The 20 classes of PASCAL VOC

aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

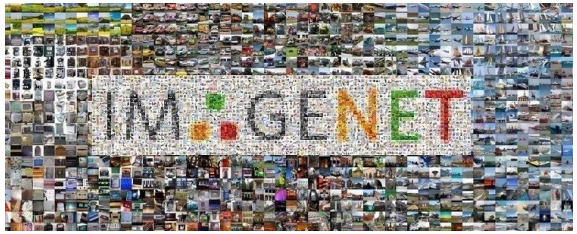


[PASCAL VOC] Everingham, Mark, et al. "**The pascal visual object classes (voc) challenge.**" IJCV, 2010.

Concept Generalization

Learn model parameters on ImageNet-1K

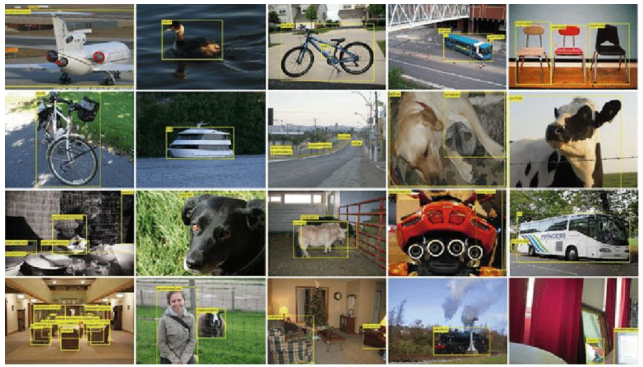
Model



Target concepts

The 20 classes of PASCAL VOC

aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

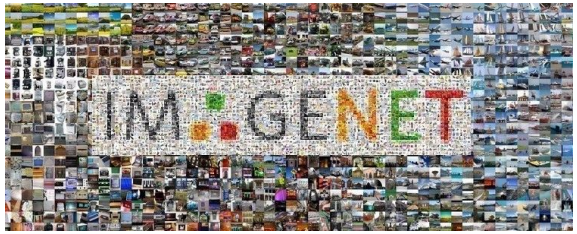


[PASCAL VOC] Everingham, Mark, et al. "**The pascal visual object classes (voc) challenge.**" IJCV, 2010.

Concept Generalization

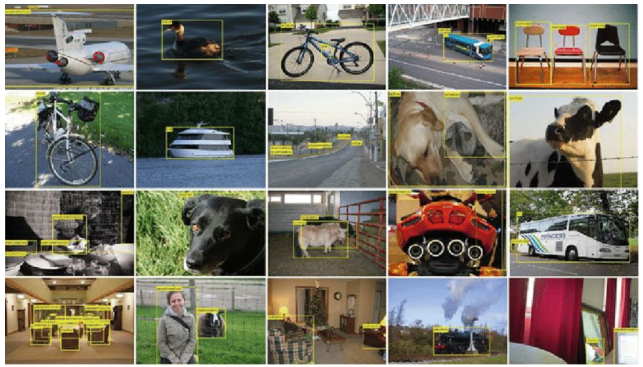
… warplane, tandem bicycle, speedboat, water bottle, minibus, school bus, trolley bus, scooter, bighorn sheep, bullet train, **television**, ox, bison, zebra …

Model



The 20 classes of PASCAL VOC

aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, **TV**

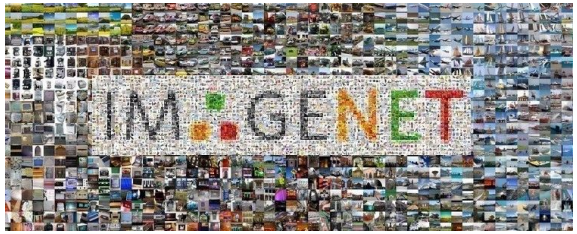


[PASCAL VOC] Everingham, Mark, et al. "**The pascal visual object classes (voc) challenge.**" IJCV, 2010.

Concept Generalization

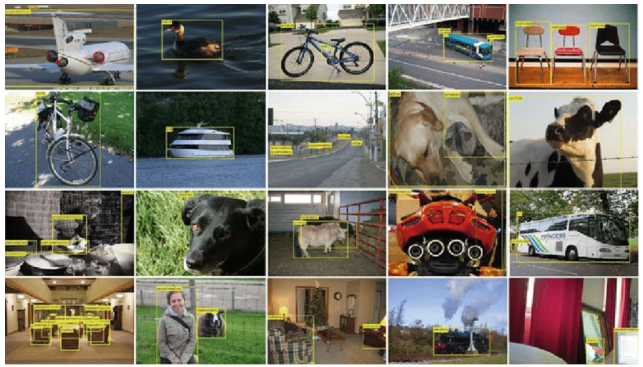
… warplane, tandem bicycle, speedboat, water bottle, **minibus**, **school bus**, **trolley bus**, scooter, bighorn sheep, bullet train, television, ox, bison, zebra …

Model



The 20 classes of PASCAL VOC

aeroplane, bicycle, boat, bottle, **bus**, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV



[PASCAL VOC] Everingham, Mark, et al. "**The pascal visual object classes (voc) challenge.**" IJCV, 2010.

Concept Generalization

… warplane, tandem bicycle, speedboat, water bottle, minibus, school bus, trolley bus, scooter, bighorn sheep, bullet train, television, **ox**, **bison**, zebra …

Model

The 20 classes of PASCAL VOC

aeroplane, bicycle, boat, bottle, bus, car, cat, chair, **cow**, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

[PASCAL VOC] Everingham, Mark, et al. "**The pascal visual object classes (voc) challenge.**" IJCV, 2010.

Ingredients for designing a better benchmark

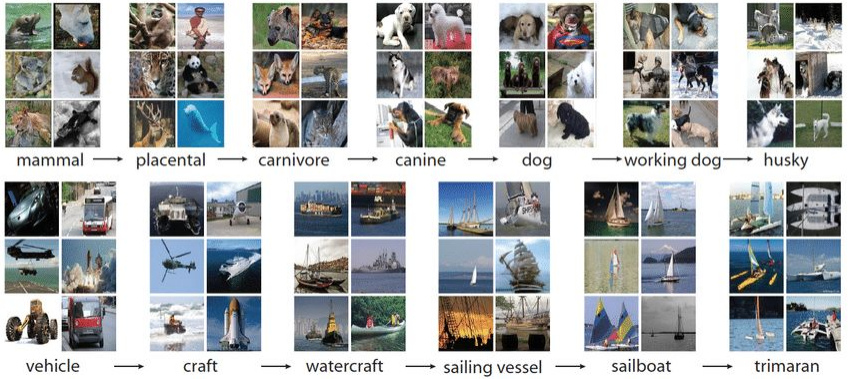
- A large set of concepts
- A controlled setup:
  - **Disjoint** set of training (seen) and testing (unseen) concepts
  - Way to measure **semantic distance** between concepts

Step 0: Pick a large dataset with many concepts

NAVER LABS

Dataset: **ImageNet-21K**
(Fall 2011 / Winter 2021 release)

- ✓ Large-scale
  >14 Million images, >21000 concepts
- ✓ Very popular training set as subset:
  LSVRC subset: [**ImageNet-1K**]
- ✓ Each concept corresponds to a synset
  from [**WordNet**]



(Figure: https://devopedia.org/imagenet)

[**ImageNet**] J. Deng, *et al* and L. Fei-Fei, **ImageNet: A Large-Scale Hierarchical Image Database.** *CVPR 2009.*
[**ImageNet-1K**] O Russakovsky, J Deng, et al. **Imagenet large scale visual recognition challenge**, IJCV, 2015.
[**WordNet**] *Miller, George A. "**WordNet: a lexical database for English.**" Communications of the ACM 38.11 (1995): 39-41.*

Step 1: Define a disjoint set of seen and unseen concepts

NAVER LABS

Seen concepts ImageNet-1K

```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark',
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
 10: 'brambling, Fringilla montifringilla',
 11: 'goldfinch, Carduelis carduelis',
 12: 'house finch, linnet',
 13: 'junco, snowbird',
 14: 'indigo bunting, indigo finch',
 15: 'robin, American robin,',
 16: 'bulbul',
 17: 'jay',
 18: 'magpie',
 19: 'chickadee',
 20: 'water ouzel, dipper',
 21: 'kite',
 22: 'bald eagle, American eagle'
 ...
 ...
```

Unseen concepts

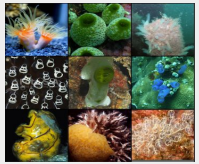
Remove/Filter:

- ✓ ImageNet-1K (seen)
- ✓ pathological concepts
- ✓ concepts with few images

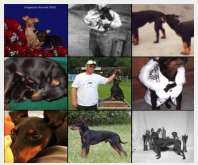
approx. **5K eligible** concepts

European wildcat

jackal



sea squirt



toy Manchester

takin

cabbageworm

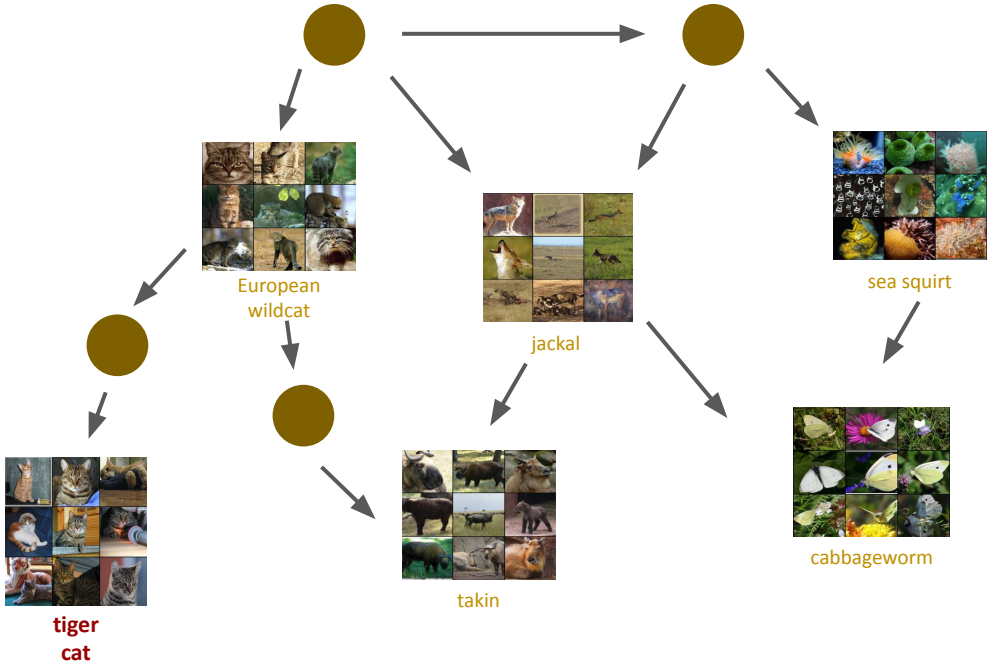
Concept-to-concept

semantic similarity:

$$\text{sim}_{\text{Lin}}(c_1, c_2) = \frac{2 \times \text{IC}(\text{LCS}(c_1, c_2))}{\text{IC}(c_1) + \text{IC}(c_2)}$$

[**Lin similarity**]

(Note: any other semantic similarity can be used, *e.g.* word2vec)

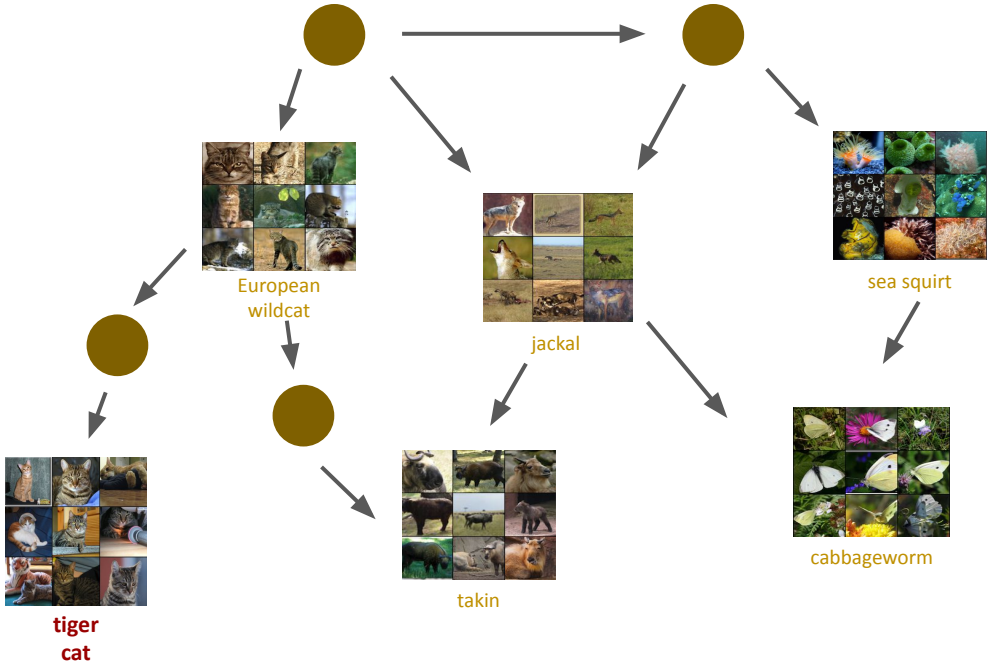


[**Lin similarity**] Dekang Lin. "**An information-theoretic definition of similarity.**" ICML 1998

Concept-set-to-concept

semantic similarity:

$$\mathcal{S}^c = \max_{c_i \in C} \text{sim}_{Lin}(c, c_i)$$



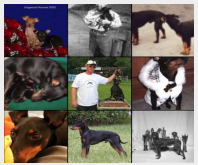
[**Lin similarity**] Dekang Lin. "**An information-theoretic definition of similarity.**" ICML 1998

NAVER LABS

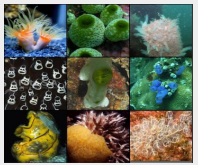
Seen concepts ImageNet-1K

```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark',
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
 10: 'brambling, Fringilla montifringilla',
 11: 'goldfinch, Carduelis carduelis',
 12: 'house finch, linnet',
 13: 'junco, snowbird',
 14: 'indigo bunting, indigo finch',
 15: 'robin, American robin,',
 16: 'bulbul',
 17: 'jay',
 18: 'magpie',
 19: 'chickadee',
 20: 'water ouzel, dipper',
 21: 'kite',
 22: 'bald eagle, American eagle'
 ...
 ...
```

Unseen concepts


toy Manchester


European wildcat


sea squirt


takin


jackal


cabbageworm

$$\forall \text{ unseen concept } c \in \mathcal{U}: \quad \mathcal{S}^c = \max_{c_i \in \text{IN-1k}} sim_{Lin}(c, c_i)$$

Step 4: Rank unseen concepts w.r.t. distance to seen

NAVER LABS

Seen concepts ImageNet-1K

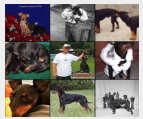
```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark',
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
 10: 'brambling, Fringilla montifringilla',
 11: 'goldfinch, Carduelis carduelis',
 12: 'house finch, linnet',
 13: 'junco, snowbird',
 14: 'indigo bunting, indigo finch',
 15: 'robin, American robin,',
 16: 'bulbul',
 17: 'jay',
 18: 'magpie',
 19: 'chickadee',
 20: 'water ouzel, dipper',
 21: 'kite',
 22: 'bald eagle, American eagle'
 ...
 ...
```

Unseen concepts

Increasing semantic distance to the **set** of seen concepts


European wildcat

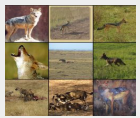
...


toy Manchester

...


takin

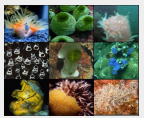
...


jackal

...


cabbageworm

...


sea squirt

Rank all unseen wrt semantic distance to **ImageNet-1K** (seen)

Step 5: Split ranked list into CoG “Levels”

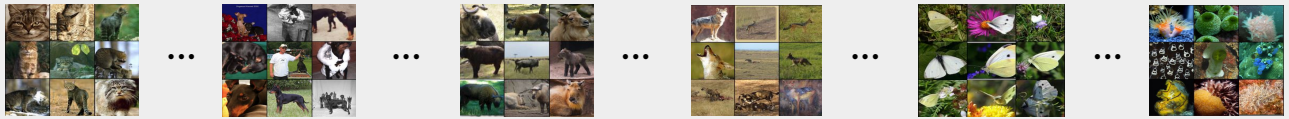
NAVER LABS

Seen concepts ImageNet-1K

```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark’,
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
 10: 'brambling, Fringilla montifringilla',
 11: 'goldfinch, Carduelis carduelis',
 12: 'house finch, linnet’,
 13: 'junco, snowbird',
 14: 'indigo bunting, indigo finch’,
 15: 'robin, American robin,’,
 16: 'bulbul',
 17: 'jay',
 18: 'magpie',
 19: 'chickadee',
 20: 'water ouzel, dipper',
 21: 'kite',
 22: 'bald eagle, American eagle’
 …
 …
```

Unseen concepts

Increasing semantic distance to the **set** of seen concepts



L_1 L_2 L_3 L_4 L_5

- Split the ranked list in five **Concept Generalization (CoG) Levels**
- Each as big as ImageNet-1K (**1000 classes**, 1.15M images)

The ImageNet-CoG Benchmark

CoG-levels: A sequence of five $L_1 \rightarrow L_5$ datasets of **unseen** concepts

...each with increasing semantic distance to the **seen** (ImageNet-1K)

Evaluation protocol:

1) **Extract features** using a model trained on ImageNet-1K (frozen)
2) **Learn linear classifiers** for ImageNet-1K and each of the five CoG levels

*We can evaluate any public ImageNet-1K pre-trained model **out-of-the-box**!*

Evaluating 30+1 recent models

ResNet50	**Baseline model** from the torchvision package (25.5M)

Architecture: Models with different backbone	
a-T2T-ViT-t-14 [73]	Visual transformer (21.1M)
a-DeiT-S [60]	Visual transformer (21.7M)
a-DeiT-S-distilled [60]	Distilled *a*-DeiT-S (21.7M)
a-Inception-v3 [55]	CNN with inception modules (25.1M)
a-NAT-M4 [38]	Neural architecture search model (7.6M)
a-EfficientNet-B1 [56]	Neural architecture search model (6.5M)
a-EfficientNet-B4 [56]	Neural architecture search model (17.5M)
a-DeiT-B-distilled [60]	Bigger version of *a*-DeiT-S-distilled (86.1M)
a-ResNet152 [25]	Bigger version of ResNet50 (58.1M)
a-VGG19 [52]	Simple CNN architecture (139.6M)

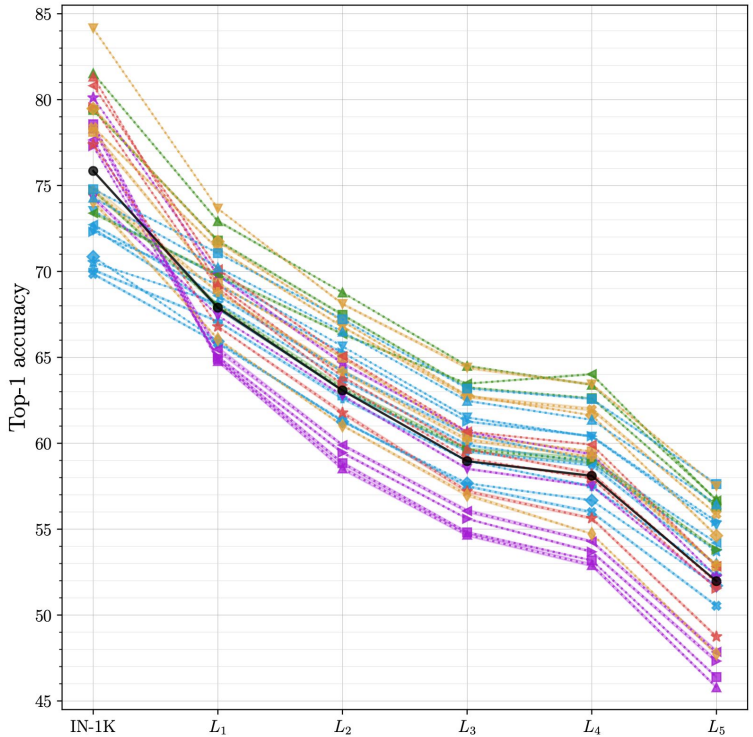
Self-supervision: ResNet50 models trained in this framework	
s-SimCLR-v2 [9, 10]	Online instance discrimination (ID)
s-MoCo-v2 [11, 24]	ID with momentum encoder and memory bank
s-BYOL [21]	Negative-free ID with momentum encoder
s-MoCHi [28]	ID with negative pair mining
s-InfoMin [58]	ID with careful positive pair selection
s-OBoW [19]	Online bag-of-visual-words prediction
s-SwAV [7]	Online clustering
s-DINO [8]	Online clustering
s-BarlowTwins [77]	Feature de-correlation using positive pairs
s-CompReSS [30]	Distilled from SimCLR-v1 [9] (with ResNet50x4)

Regularization: *ResNet50 models with additional regularization*	
r-MixUp	Label-associated data augmentation
r-Manifold-MixUp	Label-associated data augmentation
r-CutMix	Label-associated data augmentation
r-ReLabel	Trained on a "multi-label" version of IN-1K
r-Adv-Robust	Adversarially robust model
r-MEAL-v2	Distilled ResNet50

Use of web data: ResNet50 models using additional data	
d-MoPro [34]	Trained on WebVision-V1 ($\sim 2\times$)
d-Semi-Sup [67]	Pretrained on YFCC-100M ($\sim 100\times$), fine-tuned on IN-1K
d-Semi-Weakly-Sup [67]	Pretrained on IG-1B ($\sim 1000\times$), fine-tuned on IN-1K
d-CLIP [45]	Trained on WebImageText ($\sim 400\times$)

Results on ImageNet-CoG

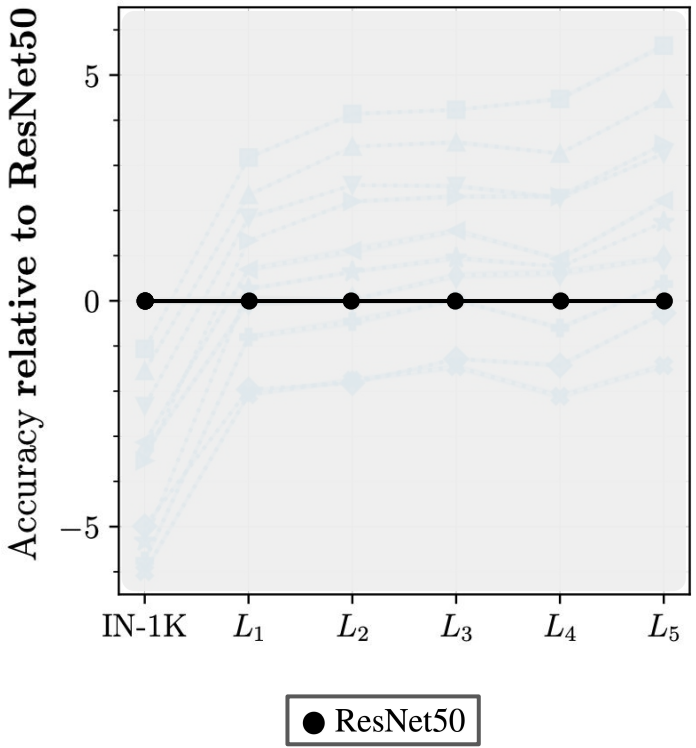
How resilient are models to the semantic distance between seen and unseen concepts?



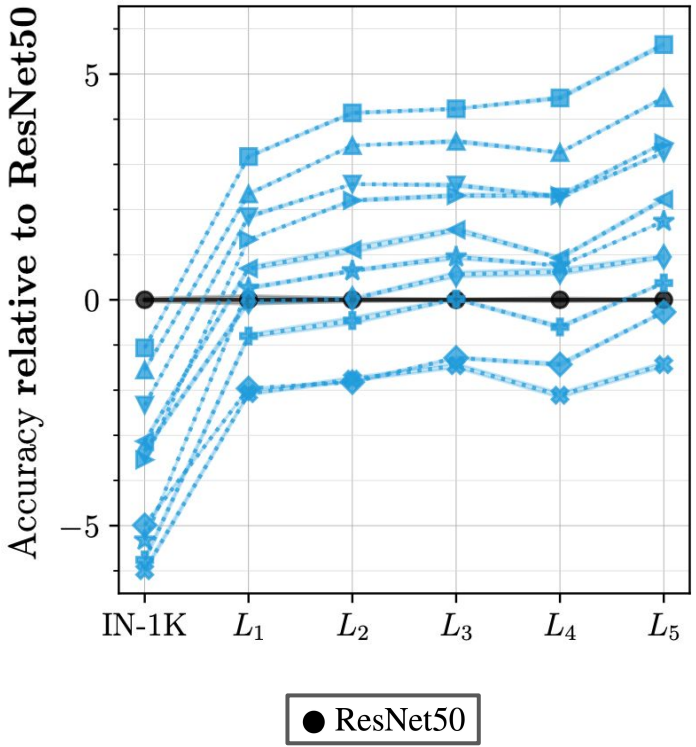
Verifying our hypothesis:

It is harder to generalize to semantically distant concepts

Results on ImageNet-CoG



Results on ImageNet-CoG

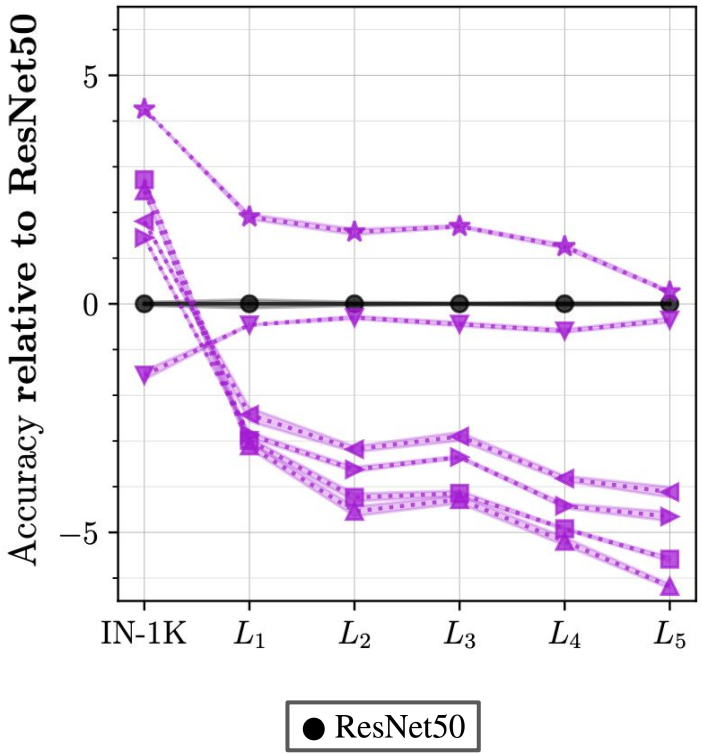


Self-supervised learning

Self-supervised models excel at concept generalization

Self-Supervision	
s-DINO	*s*-SimCLR-v2
s-SwAV	*s*-MoCo-v2
s-BarlowTwins	*s*-MoCHi
s-OBoW	*s*-CompReSS
s-BYOL	*s*-InfoMin

Results on ImageNet-CoG

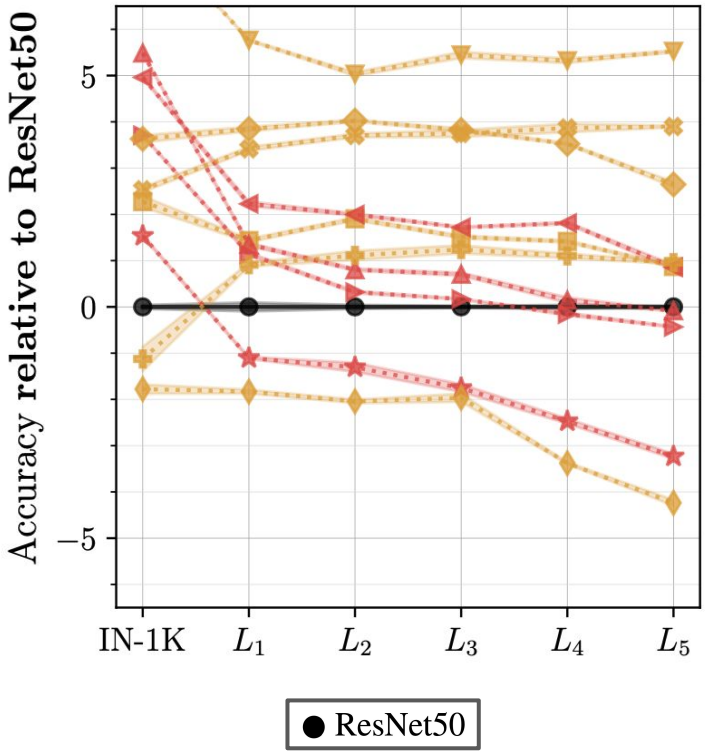


Regularization

- Model distillation generally improves CoG performance.
- Label-associated augmentation techniques deteriorate CoG performance.

Regularization	
■ *r*-ReLabel	▼ *r*-Adv-Robust
▲ *r*-CutMix	★ *r*-MEAL-v2
► *r*-MixUp	
◄ *r*-Manifold-MixUp	

Results on ImageNet-CoG

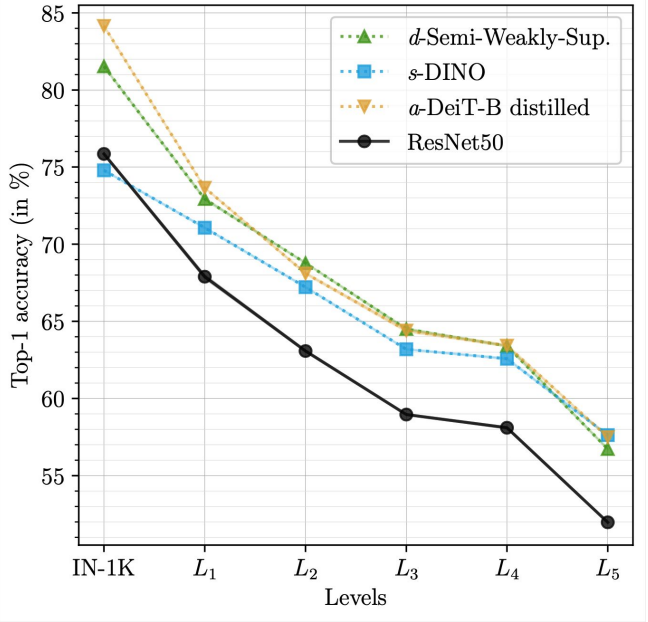


Architecture

- Visual transformers overfit more to seen concepts
- Neural architecture search seems promising for concept generalization

ResNet	Transformer	NAS & Other
● ResNet50 (23.5M)	▲ *a*-T2T-ViT-t-14 (21.1M)	★ *a*-Inception-v3 (25.1M)
■ *a*-ResNet152 (58.1M)	► *a*-DeiT-S (21.7M)	✚ *a*-EfficientNet-B1 (6.5M)
	◄ *a*-DeiT-S-distilled (21.7M)	✖ *a*-EfficientNet-B4 (17.5M)
	▼ *a*-DeiT-B-distilled (86.1M)	◆ *a*-NAT-M4 (7.6M)
		♦ *a*-VGG19 (139.6M)

Results on ImageNet-CoG



What are the top-performing models overall for concept generalization?

- Models with better and larger architectures
- Distillation
- Self-supervised models
- Models pre-trained with additional data

Take home message

- ImageNet-CoG: a new benchmark
  - Enables measuring Concept Generalization in a controlled way
  - Sequence of “levels” of unseen concepts (from ImageNet-21K) with increasing semantic distance to the seen (ImageNet-1K)
  - Analysis of many recent methods (out-of-the-box)
  - Easy to **test your ImageNet-1K model**:

https://github.com/naver/cog

MB Sariyildiz, Y. Kalantidis, D. Larlus, K. Alahari.
“Concept generalization in visual representation learning.”
ICCV 2021

Overview of this talk

Part 1

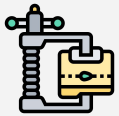
How can we improve the transfer learning performance of contrastive SSL?

MoCHi
Neurips 2020

Part 2

Can we use recent visual SSL frameworks for dimensionality reduction?

TLDR
Arxiv 2021



Part 3

How can we measure concept generalization in a more principled way?

ImageNet-CoG
ICCV 2021

[**MoCHi**] Kalantidis *et al.* "Hard negative mixing for contrastive learning." NeurIPS 2020.
[**TLDR**] Kalantidis *et al.* "TLDR: Twin Learning for dimensionality reduction" arXiv 2021.
[**ImageNet-CoG**] Sariyildiz, Kalantidis *et al.* "Concept Generalization in Visual Representation Learning" ICCV 2021.

Work with amazing co-authors!

Diane Larlus

Mert Bulent Sariyildiz

Carlos Lassance

Jon Almazan

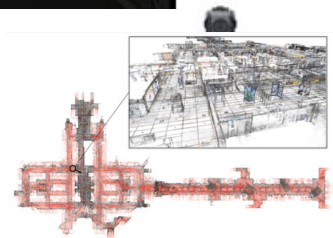
Noe Pion

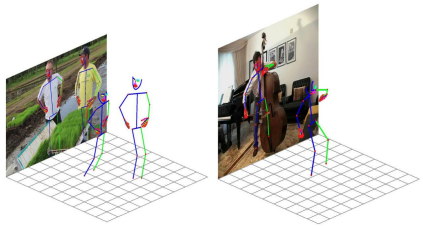
Philippe Weinzaepfel

Karteek Alahari (Inria)

Teams focusing on CV and 3D Vision

- 33 researchers/postdocs/PhDs/engineers
- 30+ top-tier publications in 2020 & 2021
  (CVPR/ECCV/NeurIPS/IJCV/ICLR/ICRA/IROS)
- Many collaborations
  - Other NLE teams
    (ML & Optimization and NLP teams)
  - Other NAVER Corp entities
    (NAVER AI Labs, CLOVA AI, NAVER LABS KOREA)
  - Academic collaborations with top-tier universities/institutes
    (University of Oxford, University of Bristol, CTU in Prague, Inria, IRI, LAAS, ENPC, SNU, MIAI Institute Grenoble)





Research scientist positions open!
Research internships possible year-round!
https://europe.naverlabs.com

MIAI Grenoble Alpes

About Yannis (and his research interests)

- Grew up in Athens, Greece
- 2009 - 2014: PhD at NTUA (Athens, supervised by *Yannis Avrithis*)
  - **Retrieval**, **clustering**, **nearest neighbor search** [ECCV12, CVPR14, ICCV15]
- 2015 - 2017: Researcher at Yahoo Research (San Francisco)
  - **Web-scale search**/classification systems [NeurIPS17 LSCVS workshop best paper]
  - **Vision and language** [IJCV17, CHI17, WSDM17, PAMI19]
- 2017 - 2019: Researcher at Facebook AI (Menlo Park)
  - **Deep architectures for vision** [ECCV18, NeurIPS18, ICCV19, CVPR19a]
  - **Video understanding/summarization** [CVPR19c, CVPR19d]
  - **Vision and language** [AAAI19, CVPR19b, ECCV20]
  - **Long-tail recognition** [ICLR20]
- 2020 - now: Researcher at NAVER LABS Europe (Grenoble)
  - **Self-supervised learning and generalization** [NeurIPS20, ICCV21]
  - Learning **expressive** visual **representations** and **adaptive** (multi-modal) **systems** [CVPR21]

Personal webpage: http://www.skamalas.com

NAVER LABS
Europe

https://europe.naverlabs.com

© NAVER LABS Corp.

https://europe.naverlabs.com

© NAVER LABS Corp.

Appendix

NAVER LABS

Predictive/Generative

- Formulated as synthesis or classification

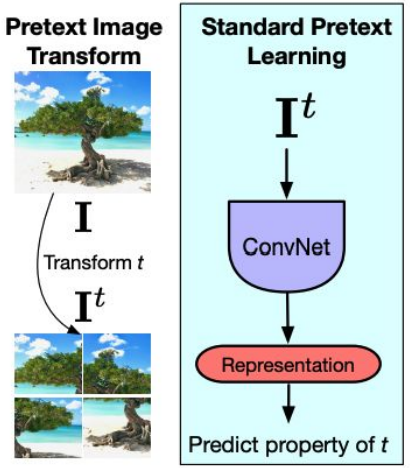


Figure from [**PIRL**]

[**PIRL**] Misra, Ishan, and Laurens van der Maaten. "Self-supervised learning of pretext-invariant representations." CVPR 2020.

Predictive/Generative

- Formulated as synthesis or classification

Contrastive

- Learning invariance to a “pretext” task

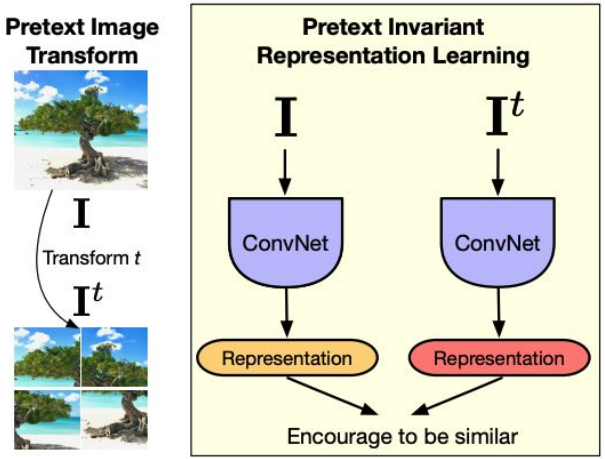


Figure from [**PIRL**]

[**PIRL**] Misra, Ishan, and Laurens van der Maaten. "Self-supervised learning of pretext-invariant representations." CVPR 2020.

Selected references

Contrastive

[**CPC**] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding." *arXiv* 2018.
[**InstDiscr**] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.
[**PIRL**] Misra, Ishan, and Laurens van der Maaten. "Self-supervised learning of pretext-invariant representations." CVPR 2020.
[**SimCLR**] Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." ICML 2020.
[**MoCo**] He, Kaiming, et al. "Momentum contrast for unsupervised visual representation learning." CVPR 2020.
[**MoCo-v2**] Chen et al. “Improved Baselines with Momentum Contrastive Learning”. arXiv preprint 2020.
[**SwAV**] Caron, Mathilde, et al. "Unsupervised learning of visual features by contrasting cluster assignments." NeurIPS 2020.
[**InfoMin Aug.**] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.
[**JCL**] Cai et al. “Joint Contrastive Learning with Infinite Possibilities”. NeurIPS 2020.
[**MoCo-v3**] Chen et al. “An Empirical Study of Training Self-Supervised Vision Transformers” arXiv preprint 2021.
[**DINO**] Caron et al. “Emerging Properties in Self-Supervised Vision Transformers”. ICCV 2021.
[**MoBy**] Xie et al. "Self-supervised learning with swin transformers." arXiv preprint 2021.
[**DirectCLR**] Li, et al. "Understanding Dimensional Collapse in Contrastive Self-supervised Learning." arXiv preprint 2021.

Non-Contrastive

[**BYOL**] Grill, Jean-Bastien, et al. "Bootstrap your own latent-a new approach to self-supervised learning." NeurIPS 2020.
[**SimSiam**] Xinlei Chen and Kaiming He “Exploring Simple Siamese Representation Learning.” CVPR 2021.
[**Barlow Twins**] Zbontar et al. “Barlow Twins: Self-Supervised Learning via Redundancy Reduction” ICML 2021.
[**DirectPred**] Tian et al. “Understanding self-supervised Learning Dynamics without Contrastive Pairs” ICML 2021.

Contrastive self-supervised learning

Positive pair: Two transformed versions of the same image

Negative: Any other image [Exemplar-CNN, InstDistr]

counter intuitive for classification!

we push representations of two images from the same class apart

[**Exemplar-CNN**] Dosovitskiy, et al. "Discriminative unsupervised feature learning with exemplar convolutional neural networks." TPAMI 2015]
[**InstDiscr**] Z Wu, Y Xiong, SX Yu, D Lin, "Unsupervised feature learning via non-parametric instance discrimination." CVPR 2018.
Cat Model: Tardar Sauce (a.k.a. grumpy cat)

Appendix: MoCHi

NAVER LABS

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1-\alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

[MoCo-v2] Chen, Xinlei, et al. "Improved baselines with momentum contrastive learning." arXiv preprint arXiv:2003.04297 (2020)

Mixing of Contrastive Hard Negatives

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1 - \alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

↓

How many of the hardest existing negatives to use?

Mixing of Contrastive Hard Negatives

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1-\alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

↓

How many points to synthesize by mixing two negatives?

Mixing of Contrastive Hard Negatives

- Feature Normalization

$$\mathbf{h}_k = \frac{\tilde{\mathbf{h}}_k}{\|\tilde{\mathbf{h}}_k\|_2}, \text{ where } \tilde{\mathbf{h}}_k = \alpha_k \mathbf{n}_i + (1-\alpha_k)\mathbf{n}_j,$$

- We run MoCHi on top of [MoCo-v2]
  - 2-layer MLP head, cosine learning rate
- MoCHi notation:

MoCHi (N, s, s')

↓

How many points to synthesize by mixing the query with a negative?

Results on ImageNet-1k and PASCAL VOC

Linear Classification
(train on ImageNet-1K without labels, than learn linear classifiers on the same set)

➢ MoCHi retains the strong performance of MoCo-v2 but shows no gains

Method	IN-1k Top1	VOC 2007 AP_{50}	VOC 2007 AP	VOC 2007 AP_{75}
	100 epoch training			
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
	200 epoch training			
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	**82.8** (±0.2) (0.3)	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
	800 epoch training			
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	**83.3** (±0.1) (0.6)	57.3 (±0.2) (0.5)	**64.2** (±0.4) (0.3)
Supervised [21]	76.1	81.3	53.5	58.8

Results on ImageNet-1k and PASCAL VOC

Transfer learning
(Train on ImageNet-1K,
fine-tune on PASCAL VOC
for Object detection)

MoCHi helps the model **learn faster**:

- ➢ Strong performance after *only 100 epochs* of pre-training
- ➢ MoCHi after 200 epochs performs similar to MoCo-v2 after 800 epochs
- ➢ Gains persist after longer training (800 epochs)

Method	IN-1k Top1	VOC 2007 AP_{50}	VOC 2007 AP	VOC 2007 AP_{75}
100 epoch training				
MoCo-v2 [10]*	63.6	80.8 (±0.2)	53.7 (±0.2)	59.1 (±0.3)
+ MoCHi (256, 512, 0)	63.9	81.1 (±0.1) (0.4)	54.3 (±0.3) (0.7)	60.2 (±0.1) (1.2)
+ MoCHi (256, 512, 256)	63.7	**81.3** (±0.1) **(0.6)**	54.6 (±0.3) (1.0)	60.7 (±0.8) (1.7)
+ MoCHi (128, 1024, 512)	63.4	81.1 (±0.1) (0.4)	**54.7** (±0.3) **(1.1)**	**60.9** (±0.1) **(1.9)**
200 epoch training				
MoCo-v2 [10]	67.7	82.4	57.0	63.6
InfoMin Aug. [39]	70.1	82.7	**57.6**	**64.6**
MoCo-v2 [10]*	67.9	82.5 (±0.2)	56.8 (±0.1)	63.3 (±0.4)
+ MoCHi (1024, 512, 256)	68.0	82.3 (±0.2) (0.2)	56.7 (±0.2) (0.1)	63.8 (±0.2) (0.5)
+ MoCHi (512, 1024, 512)	67.6	82.7 (±0.1) (0.2)	57.1 (±0.1) (0.3)	64.1 (±0.3) (0.8)
+ MoCHi (256, 512, 0)	67.7	**82.8** (±0.2) (0.3)	57.3 (±0.2) (0.5)	64.1 (±0.1) (0.8)
800 epoch training				
SvAV [7]	*75.3*	82.6	56.1	62.7
MoCo-v2 [10]	71.1	82.5	**57.4**	64.0
MoCo-v2[10]*	69.0	82.7 (±0.1)	56.8 (±0.2)	63.9 (±0.7)
+ MoCHi (128, 1024, 512)	68.7	**83.3** (±0.1) (0.6)	57.3 (±0.2) (0.5)	**64.2** (±0.4) (0.3)
Supervised [21]	76.1	81.3	53.5	58.8

Results on COCO

	Object Detection			Instance Segmentation		
Pre-train	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised [13]	38.2	58.2	41.6	33.3	54.7	35.2
	100 epoch pre-training					
MoCo-v2 [6]	37.0 (±0.1)	56.5 (±0.3)	39.8 (±0.1)	32.7 (±0.1)	53.3 (±0.2)	34.3 (±0.1)
+ MoCHi (256, 512, 0)	37.5 (±0.1) (↑0.5)	57.0 (±0.1) (↑0.5)	40.5 (±0.2) (↑0.7)	33.0 (±0.1) (↑0.3)	53.9 (±0.2) (↑0.6)	34.9 (±0.1) (↑0.6)
+ MoCHi (128, 1024, 512)	**37.8** (±0.1) **(↑0.8)**	**57.2** (±0.0) **(↑0.7)**	**40.8** (±0.2) **(↑1.0)**	**33.2** (±0.0) **(↑0.5)**	54.0 (±0.2) **(↑0.7)**	35.4 (±0.1) **(↑1.1)**
	200 epoch pre-training					
MoCo [13]	38.5	58.3	41.6	33.6	54.8	35.6
MoCo (1B image train) [13]	39.1	58.7	42.2	34.1	55.4	36.4
InfoMin Aug. [28]	39.0	58.5	42.0	34.1	55.2	36.3
MoCo-v2 [6]	39.0 (±0.1)	58.6 (±0.1)	41.9(±0.3)	34.2 (±0.1)	55.4 (±0.1)	36.2 (±0.2)
+ MoCHi (256, 512, 0)	39.2 (±0.1) (↑0.2)	58.8 (±0.1) (↑0.2)	42.4 (±0.2) (↑0.5)	34.4 (±0.1) (↑0.2)	55.6 (±0.1) (↑0.2)	36.7 (±0.1) (↑0.5)
+ MoCHi (128, 1024, 512)	39.2 (±0.1) (↑0.2)	58.9 (±0.2) (↑0.3)	42.4 (±0.3) (↑0.5)	34.3 (±0.1) (↑0.2)	55.5 (±0.1) (↑0.1)	36.6 (±0.1) (↑0.4)
+ MoCHi (512, 1024, 512)	**39.4** (±0.1) **(↑0.4)**	**59.0** (±0.1) **(↑0.4)**	**42.7** (±0.1) **(↑0.8)**	**34.5** (±0.0) **(↑0.3)**	**55.7** (±0.2) **(↑0.3)**	**36.7** (±0.1) **(↑0.5)**

Gains also consistent on COCO:

- MoCHi outperforms recent methods like [InfoMin Aug]

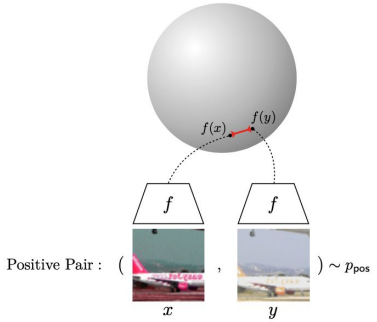
[InfoMin Aug.] Tian, Yonglong, et al. "What makes for good views for contrastive learning." NeurIPS 2020.

Uniformity and alignment scores [Wang & Isola]

NAVER LABS

Alignment

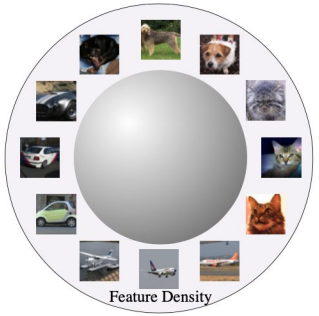
- Average distance between representations with the same class



Alignment: Similar samples have similar features.

Uniformity

- Average pairwise distance between all embeddings



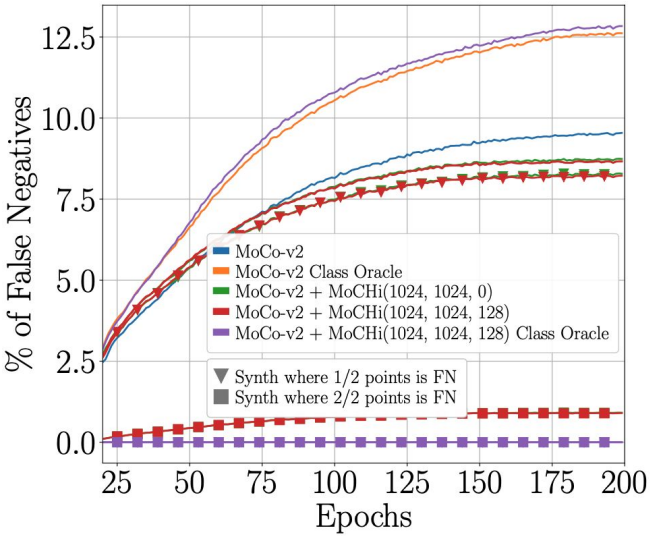
Uniformity: Preserve maximal information.

[Wang & Isola] Wang, Tongzhou, and Phillip Isola. "Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere." ICML 2020.

Analysis using a class label “oracle”

False Negatives (FN): Use ImageNet labels to measure negative items that are:

- from the same class as the **q**
- Highly rank wrt logits, *i.e.* in the top-1024 highest logits for **q**



Observations when looking at FN across epochs:

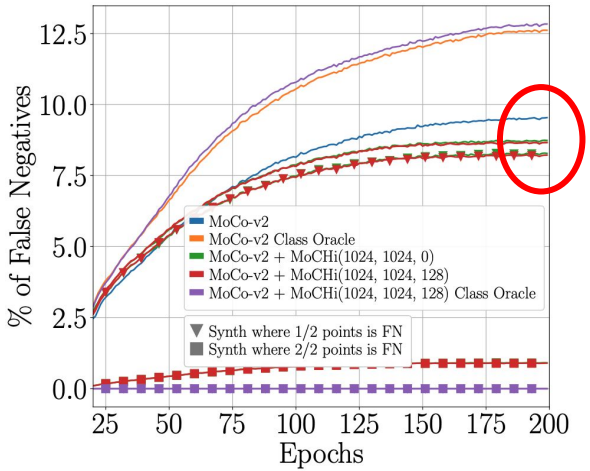
- FN in the top-k increase with training
- Only a small percentage (~1%) of the points synthesized with MoCHi are *definitely* FN
- MoCHi has overall a smaller percentage of false negatives than MoCo

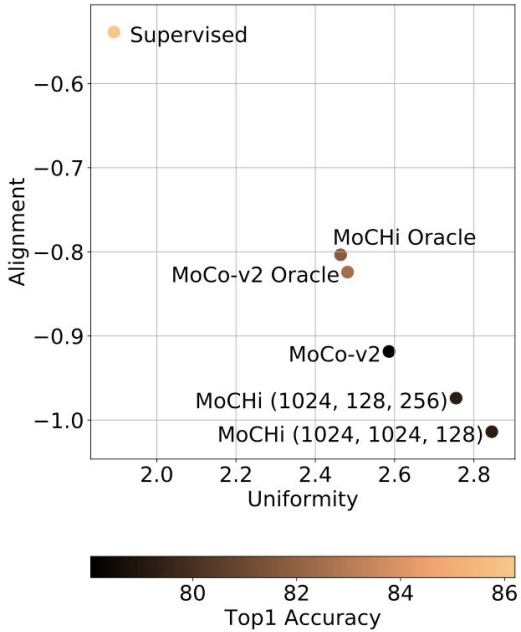
Why does MoCHi perform better for downstream tasks?

Alignment

Supervised > MoCo > MoCHi

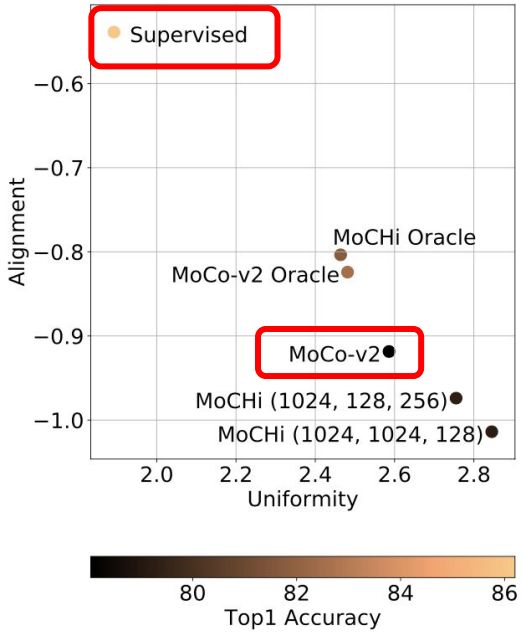
This result confirms the plot





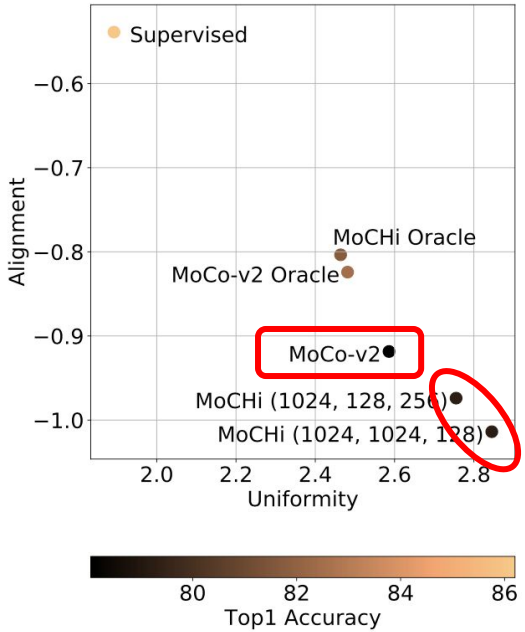
Utilization of the embedding space

- Contrastive SSL (MoCo) utilizes the embedding space “more” than training with Cross Entropy (supervised)



Utilization of the embedding space

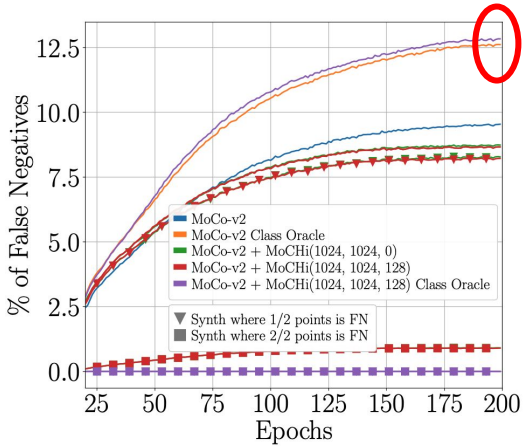
- Contrastive SSL (MoCo) utilizes the embedding space “more” than training with Cross Entropy (supervised)
- Adding synthetic hard negative (MoCHi) results in utilizing the space even more!

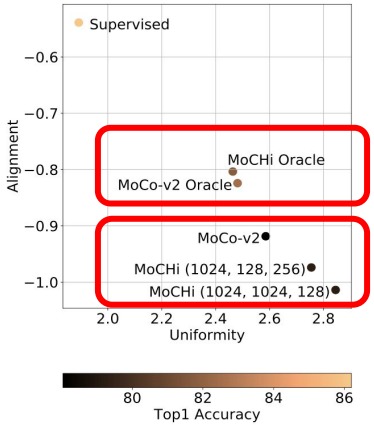


“Oracle” runs

What if we didn’t have **FN**?

- Upper bound: simply discard images with the same label as th negatives
- Oracle runs show:
  - higher percentage of FN
  - higher alignment score





“Oracle” runs

What if we didn’t have **FN**?

- Upper bound: simply discard images with the negatives
- Oracle runs show:
  - higher percentage of FN
  - higher alignment score
- Performance:
  - Closing the gap with supervised

Linear classification accuracy (ImageNet-100)

	Using Class Oracle
MoCo-v2* (200 epochs)	81.8
+ MoCHi (1024, 1024, 128) (200 epochs)	82.5
+ MoCHi (1024, 1024, 128) (400 epochs)	84.2
+ MoCHi (1024, 1024, 128) (800 epochs)	85.2
Cross-entropy classification (supervised)	86.2

	Acc | AP-50		AP	AP-75
	Using Class Oracle			
Cross-entropy classification (supervised)	76.1	81.3	53.5	58.8
MoCo-v2 [10] + MoCHi (512, 1024, 512)	72.6	*83.3*	*57.7*	*64.6*

ImageNet-1K PASCAL VOC

see also: Khosla, Prannay, et al. "Supervised contrastive learning." NeurIPS 2020.

Appendix: TLDR

NAVER LABS

Datasets and tasks of the main paper

Task (Metric)	Input feature space	Dimensionality reduction dataset	Test dataset
Landmark Retrieval (mAP)	ResNet50 features $D = 2048$ trained on Landmarks-clean (40k) [3, 17]	Google Landmarks [55] (1.5M)	ROxford5K (5k) RParis6K (6k) [40]
Argument Retrieval (Recall@100)	BERT Features $D = 768$ trained on MSMarco (8.8M) [36]	Webis-Touché 2020 (380k) [6]	ArguAna (3k) [54]

Table 1: **Datasets and tasks** of the main paper.

Methods compared for Image Retrieval

Method	(Self-) supervision	Encoder	Projector	Loss	Notes
PCA [38]	unsupervised	linear	linear	Reconstruction MSE + orthogonality	Used for dimensionality reduction in SoTA methods like DELF, GeM, GeM-AP and HOW
DrLim	neighbor-supervised	MLP	None	Contrastive	[20] (*very low performance*)
Contrastive	neighbor-supervised	linear	MLP	Contrastive	[20] with projector
MSE	unsupervised	linear	MLP	Reconstruction MSE	TLDR with MSE loss
TLDR$_{\mathcal{G}}$	denoising	linear	MLP	Barlow Twins	TLDR with noise as distortion
TLDR	neighbor-supervised	linear	MLP	Barlow Twins	
TLDR$_{1,2}$	neighbor-supervised	fact. linear	MLP	Barlow Twins	
TLDR$^{\star}_{1,2}$	neighbor-supervised	MLP	MLP	Barlow Twins	

Table 2: **Compared Methods.** For *unsupervised* methods the objective is based on reconstruction, *neighbor-supervised* methods utilize nearest neighbors as pseudo-labels to learn, *denoising* learns to ignore added Gaussian noise.

TLDR: Landmark image retrieval results

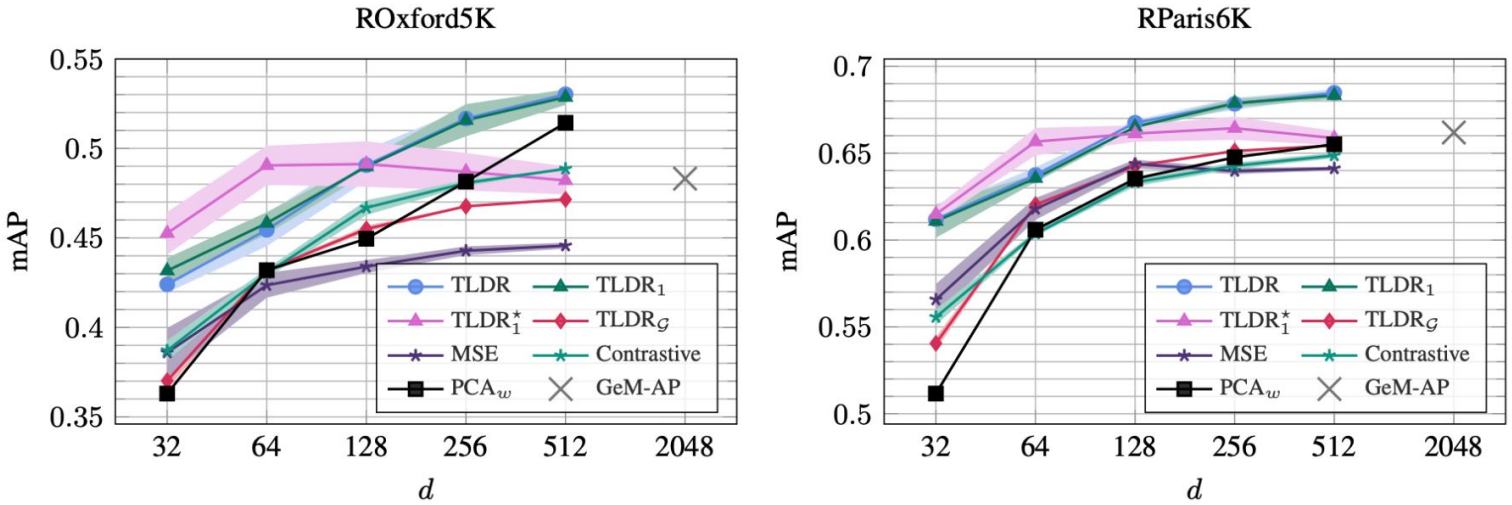


Figure 2: **Image retrieval experiments**. Mean average precision (mAP) on ROxford5K (left) and RParis6K (right) as a function of the output dimensions d. We report TLDR with different encoders: linear (TLDR), factorized linear with 1 hidden layer (TLDR$_1$), and a MLP with 1 hidden layer (TLDR$_1^\star$), the projector remains the same (MLP with 2 hidden layers). We compare with PCA with whitening, two baselines based on TLDR, but which respectively train with a reconstruction (MSE) and a contrastive (Contrastive) loss, and also with *TLDR*$_\mathcal{G}$, a variant of TLDR which uses Gaussian noise to synthesize pairs. The original GeM-AP performance is also reported.

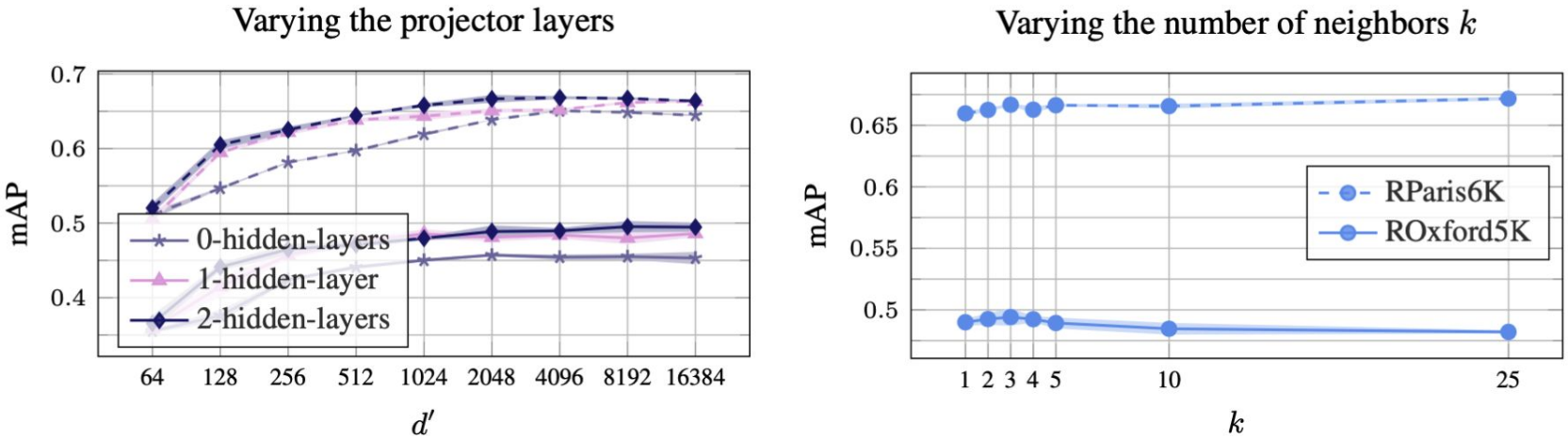
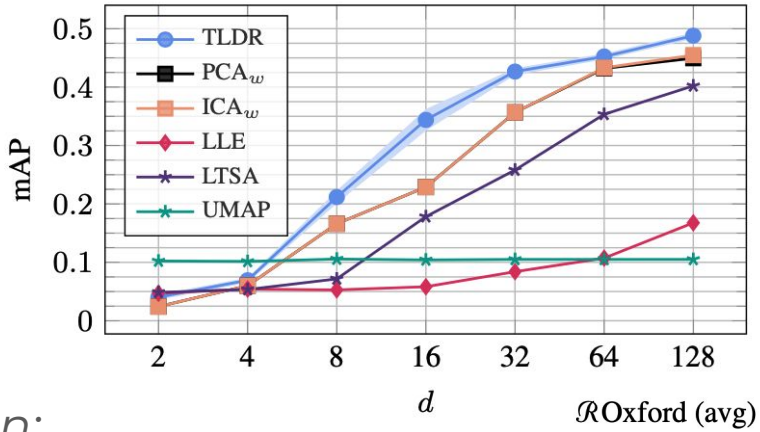


Figure 4: **Impact of TLDR hyper-parameters** with a linear encoder and $d = 128$. Dashed (solid) lines are for RParis6K-Mean (ROxford5K-Mean). (Left) Impact of the *auxiliary* dimension d' and the number of hidden layers in the projector. (Right) Impact of the number of neighbors k. We see how the algorithm is robust to the number of neighbors used.

Ablations and discussion

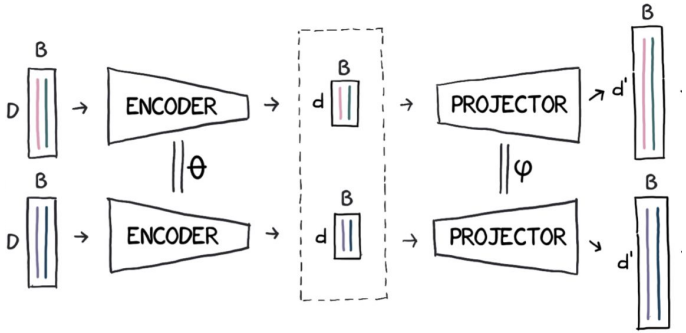
Compared to manifold methods in d < 32:

- **TDLR** outperforms all for d > 4
- Note: Linear dimensionality reduction is a very strong baseline for d > 8

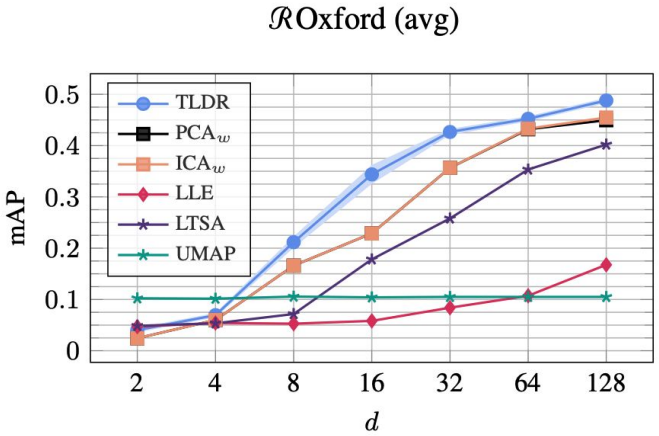


Highlights of TLDR hyper-parameter ablation:

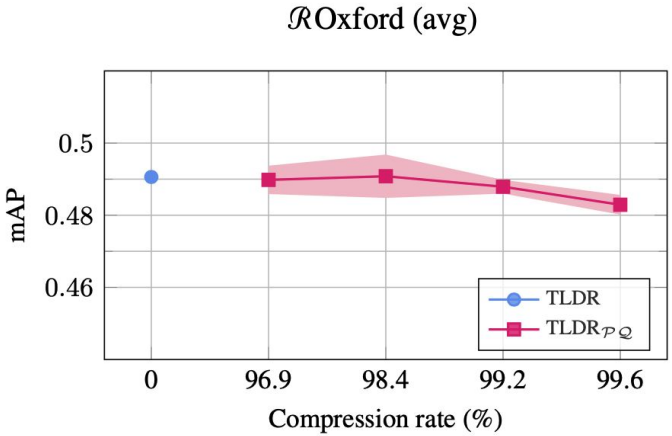
- High loss (projector output) dimension helps
- TLDR is robust to approximate k-NN (no perf. loss for up to 99% compression)
- TLDR is robust w.r.t. **k** hyper-parameter



Comparisons to manifold learning & a-NN

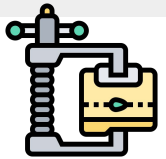


(a) Comparisons to manifold learning methods.



(b) Effect of approximate nearest neighbors.

Twin Learning for Dimensionality Reduction



Compared to manifold learning methods:

- Trivially scalable w.r.t. dataset size
- Trivial **out-of-sample** generalization
- **Higher performance** than all methods tested for d > 8

Compared to linear dimensionality reduction (eg PCA/ICA)*:*

- Identical encoding speed (linear encoder)
- **Higher performance** than the best linear variant on most tasks we tested

Appendix: CoG

NAVER LABS

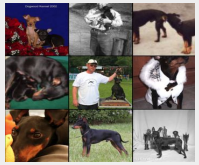
Seen concepts: ImageNet-1K

```
{0: 'tench, Tinca tinca',
 1: 'goldfish, Carassius auratus',
 2: 'great white shark, white shark',
 3: 'tiger shark, Galeocerdo cuvieri',
 4: 'hammerhead, hammerhead shark',
 5: 'electric ray, crampfish, numbfish',
 6: 'stingray',
 7: 'cock',
 8: 'hen',
 9: 'ostrich, Struthio camelus',
 10: 'brambling, Fringilla montifringilla',
 11: 'goldfinch, Carduelis carduelis',
 12: 'house finch, linnet',
 13: 'junco, snowbird',
 14: 'indigo bunting, indigo finch',
 15: 'robin, American robin,',
 16: 'bulbul',
 17: 'jay'
 18: 'magp
 19: 'chic
 20: 'wate
 21: 'kite
 22: 'balc
 ...
 ...
```

Unseen concepts: filtered ImageNet-21K

Increasing semantic distance to **set** of seen concepts

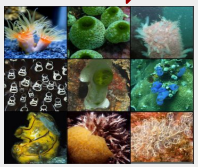

European wildcat


toy Manchester


takin


jackal


cabbageworm

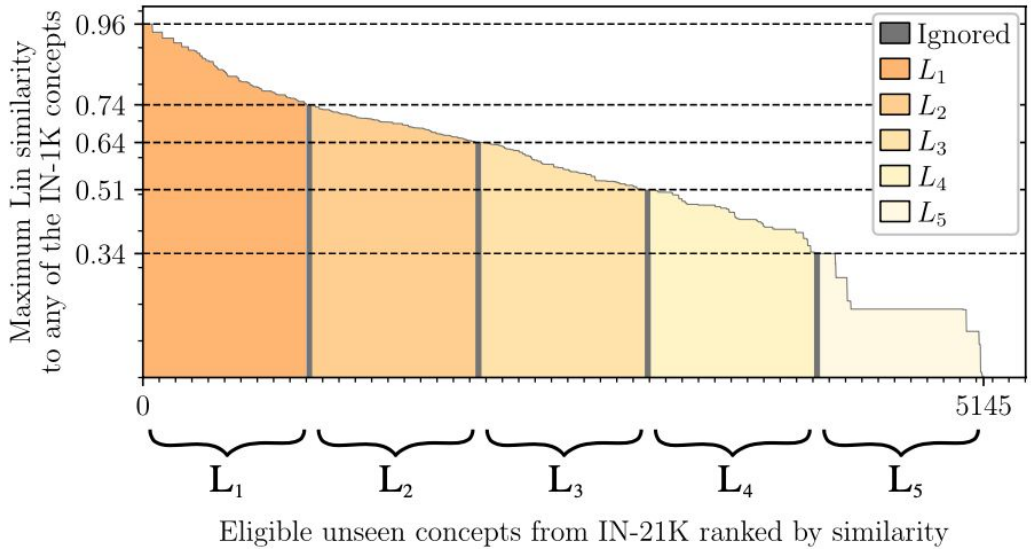

sea squirt

Rank them wrt their semantic distance to ImageNet-1K

- *Lin similarity* [Lin] over the WordNet graph
- Use concept-to-set Lin similarity

[Lin] Dekang Lin. "**An information-theoretic definition of similarity.**" ICML 1998

The ImageNet-CoG Benchmark



Nearest neighbor per Level - “orange”

n07747607 (IN-1K): orange

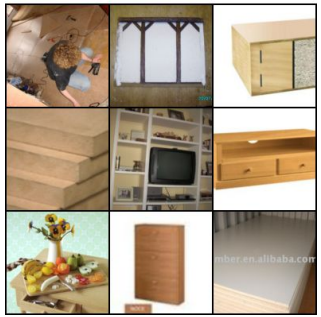
n07749312 (Level-1): Valencia orange

n07767171 (Level-2): mulberry

n07766173 (Level-3): litchi

n12158031 (Level-4): gourd

n15102455 (Level-5): chipboard



Concept generalization benchmark wishlist

We would ideally have:

- Disjoint set of **seen** (training) and **unseen** (target) concepts
- Same image and class statistics
  - overall size, # concepts, # images per class
  - Same domain/sampling strategy of images
- Same annotation process and similar label noise
- Known “semantic distance” between seen and target concepts
- Bonus: it should be easy for researchers to evaluate

possible if using a single large dataset for training and transfer

possible if there is a concept ontology

possible if we use a popular training set

Can we design a benchmark that would satisfy all these?

Fine-tuning and feature dimensions

Model	Feature Dim.
All models with ResNet [25] backbone	2048
a-T2T-ViT-t-14 [73]	384
a-DeiT-S [60]	384
a-DeiT-B-distilled [60]	768
a-NAT-M4 [38]	1536
a-EfficientNet-B1 [56]	1280
a-EfficientNet-B4 [56]	1792
a-VGG19 [52]	4096

Table 3: Unique architectures used by the models we evaluate in Sec. 4 of the main paper and the dimensionality of the feature vectors we extract from these architectures.

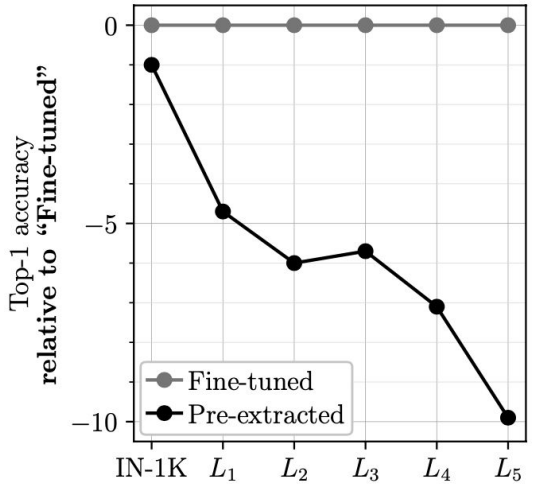


Figure 6: Comparison of training linear classifiers on **pre-extracted features** *vs.* **fine-tuning** backbones on each level. Y-axis shows the top-1 accuracies obtained **relative** to the accuracy of the fine-tuned models.

Label noise in ImageNet-CoG

	Concepts in L_5		**Concepts in IN-1K**	
				
Ground Truth	Rock climbing	Handball	Ptarmigan	Dalmatian
	Predictions by IN-1K classifier		**Predictions by L_5 classifier**	
Predicted labels	Cliff	Soccer ball	Peak	Snow

word2vec as semantic distance

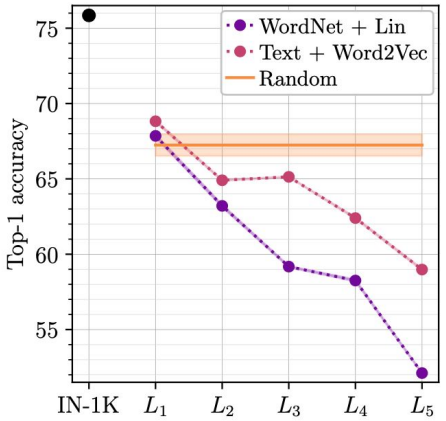
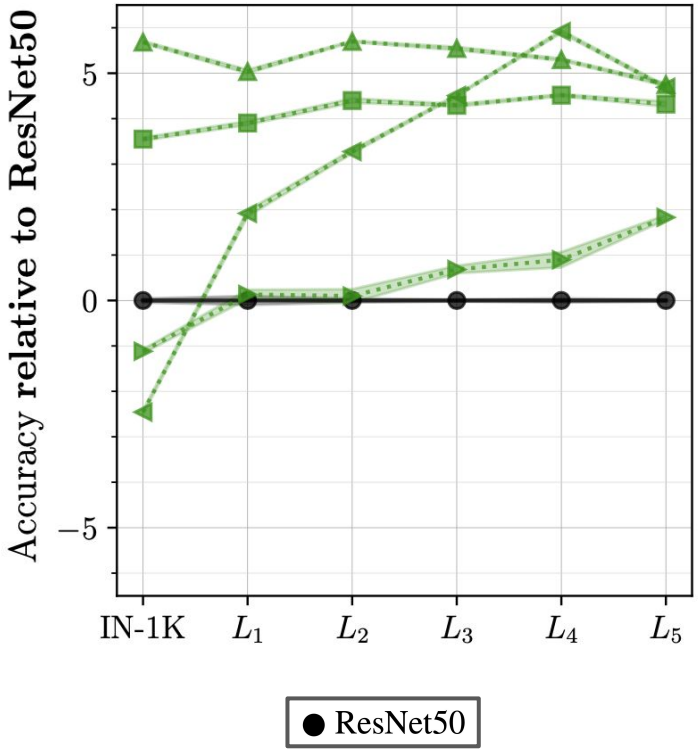


Figure 7: **Semantic similarities** of the concepts captured by (i) Lin similarity [36] on WordNet graph [43] and (ii) cosine similarity of word2vec embeddings [68] extracted from textual descriptions of concepts, *vs*. **visual similarities** encoded by ResNet50, on IN-1K and generalization levels $L_{1/2/3/4/5}$ of ImageNet-CoG. We report the performance of linear logistic regression classifiers trained on features extracted from the global average pooling layer of ResNet50. The orange line shows results obtained on 1000 *random* unseen concepts (line represents the mean accuracy obtained over 15 random splits).

Generalization to unseen concepts

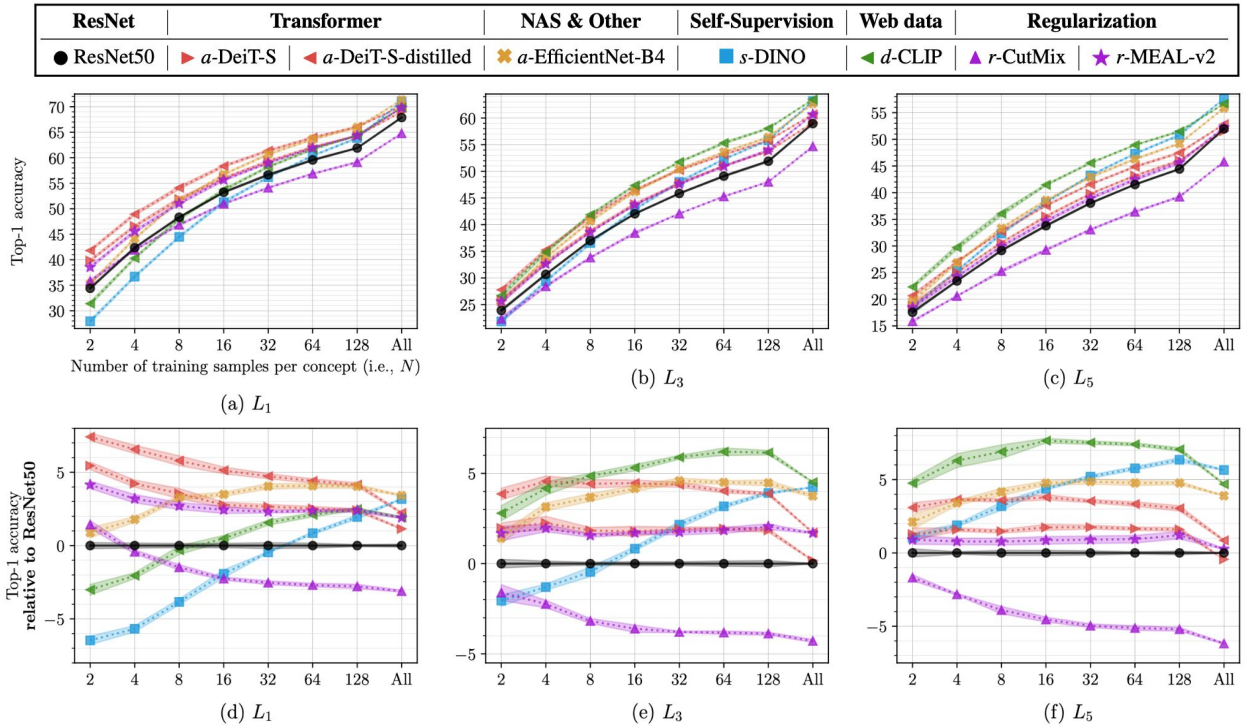


Additional noisy (web) data

Using noisy web data highly improves concept generalization

Note:
CLIP model comparison is unfair

Generalization from a few samples per concept



- Transformer-based models are strong few-shot learners
- Model Distillation and Neural Architecture Transfer exhibit consistent gains
- Bigger models and additional web data help at few shot learning